

МІНІСТЕРСТВО ОСВІТИ І НАУКИ,
МОЛОДІ ТА СПОРТУ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ
«ХАРКІВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

МЕТОДИЧНІ ВКАЗІВКИ
для проведення лабораторних занять з курсу
«Програмне забезпечення мікропроцесорних систем»

для студентів напряму підготовки 050202
«Автоматизація та комп'ютерно-інтегровані технології»

Затверджено
редакційно-видавничою
радою університету,
протокол № 2 від 07.12.2011 р.

Харків
НТУ «ХПІ»
2012

Методичні вказівки для проведення лабораторних занять з курсу «Програмне забезпечення мікропроцесорних систем» для студентів на-пряму підготовки 050202 «Автоматизація та комп'ютерно-інтегровані технології» денної та заочної форм навчання / Уклад.: В. І. Тошинський, І. Г. Лисаченко, І. І. Литвиненко та ін. – Х.: НТУ «ХПІ», 2012. – 56 с.

Укладачі: В. І. Тошинський
І. Г. Лисаченко
І. І. Литвиненко
О. Г. Шутинський
О. М. Дзевочко
Е. Є. Герман

Рецензент: В. В. Скородєлов, доц. канд. техн. наук, проф. каф.
обчислювальної техніки та програмування

Кафедра автоматизації хіміко-технологічних систем
та екологічного моніторингу

ВСТУП

Дані методичні вказівки розроблені для проведення лабораторних робіт зі студентами денної та заочної форм навчання за напрямом «Автоматизація та комп'ютерно-інтегровані технології» з дисципліни «Програмне забезпечення мікропроцесорних систем». Вони вміщують необхідні теоретичні відомості для розробки прикладного програмного забезпечення (ППЗ) для систем управління хіміко-технологічними процесами.

Для розробки та завантаження прикладного програмного забезпечення до складу програмно-технічного комплексу входить ПК з встановленим спеціальним програмним забезпеченням (СПЗ) – комплексом програмування контролерів *CoDeSys V2.3* [1, 2] (скорочення від *Controllers Development System*), який розроблений компанією *3S - Smart Software Solutions GmbH* (Німеччина) [5]. Опис основних принципів роботи з контролерами виробництва ОВЕН наведено в роботах [3, 4], а технічну підтримку можна одержати за адресою в *Internet* [6].

Лабораторні роботи проводяться на стендах, що мають у своєму складі програмований логічний контролер (ПЛК), об'єкт управління (модель теплообмінника) та дискретні і аналогові датчики. Модель теплообмінника складається з нагрівача (нагрівальний резистор) та охолоджувача (вентилятор обдування). Контрольованим параметром є температура повітря всередині теплообмінника, яка вимірюється за допомогою або термометра опору, або термоелектричного перетворювача – термопари. Також на стенді присутній імітатор датчика положення засувки – змінний резистор. Для імітування сигналів управління та дискретних датчиків до ПЛК підключений емулятор дискретних сигналів з перемикачами типу «сухий контакт».

Метою проведення лабораторних робіт є закріплення теоретичних знань та отримання практичних навичок програмування ПЛК при створенні систем управління хіміко-технологічними процесами. Лабораторні роботи розроблені за принципом поступового ускладнення завдань: від звичайного логічного управління до реалізації складних законів регулювання (ПД- або двопозиційного закону регулювання) технологічного параметра, які здійснюють ПЛК. Також в лабораторних робо-

тах вивчається метод аналогового управління дискретним вихідним елементом, так зване ШІМ-регулювання.

Обладнання всіх стендів для виконання лабораторних робіт вироблено компанією OBEH (Росія), яка надала його безкоштовно, згідно з програмою підтримки вищих навчальних закладів.

Лабораторна робота №1

ЗАГАЛЬНІ ПРИНЦИПИ ПРОГРАМУВАННЯ КОНТРОЛЕРІВ **OBEH** В СЕРЕДОВИЩІ **CODESYS**

1.1. Мета роботи

– ознайомлення з обладнанням стенда, вивчення його принципової та структурної схем;

– закріплення теоретичних знань щодо побудови програмованих логічних контролерів **OBEH** серій *ПЛК100/150/154* та структури програмного комплексу *CoDeSys V2.3*;

– отримання практичних навичок та прийомів роботи в середовищі *CoDeSys V2.3* в процесі створення прикладного програмного забезпечення для ПЛК: *створення проекту, конфігурування контролера, написання програми користувача, компіляція та емуляція роботи програми, завантаження проекту до ПЛК та налагодження його.*

1.2. Опис стенда

Загальний вигляд стенда наведено на рис. 1.1. На цьому рисунку виносками позначені: 1 – імітатор нагрівача, 2 – імітатор охолоджувача, 3 – клавіша включення панелі оператора (ПО), 4 – імітатор аналогового сигналу ($R\sim$), 5 – ПЛК, 6, 7 – імітатори дискретних сигналів, 8 – ПО, 9 – клавіша включення ПЛК, 10 – датчик температури. Нижче наведено опис елементів стенда:

1. Імітатор нагрівача – потужний дрововий опір в керамічному корпусі типу *ПЭВ-100* номіналом 750 Ом.

2. Імітатор охолоджувача – вентилятор обдування з живленням постійної напруги.

3. Клавіша вмикання живлення ПО.
4. Імітатор аналогового сигналу – змінний опір номіналом 0...1 кОм.
5. Контролер *ОВЕН* моделі *ПЛК150-220.И-L* з внутрішнім джерелом живлення, який є моноблоком, що об'єднує в собі контактні групи для підключення дискретних та аналогових сигналів вводу/виводу, а також інтерфейси обміну: *RS-232* – для завантаження програм та обміну даними з ПК; *RS-485* – для мережного обміну з іншими пристроями; *Ethernet* – для завантаження програм та обміну даними з ПК.
6. Імітатор входних дискретних сигналів *ЭДИ-6* для ПЛК – шість перемикачів типу «сухий контакт».
7. Імітатор входних дискретних сигналів для ПО – чотири перемикачі типу «сухий контакт».
8. Панель оператора *ОВЕН СМІІ-1-220* для індикації даних з функціями редагування і для роботи в мережах *RS-485* та *RS-232* за протоколами *ModBus ASCII/RTU* та *ОВЕН*.
9. Клавіша вмикання живлення ПЛК.
10. Датчик температури – термоелектричний перетворювач (термопара). Тип та модель датчика вказані на бирці.

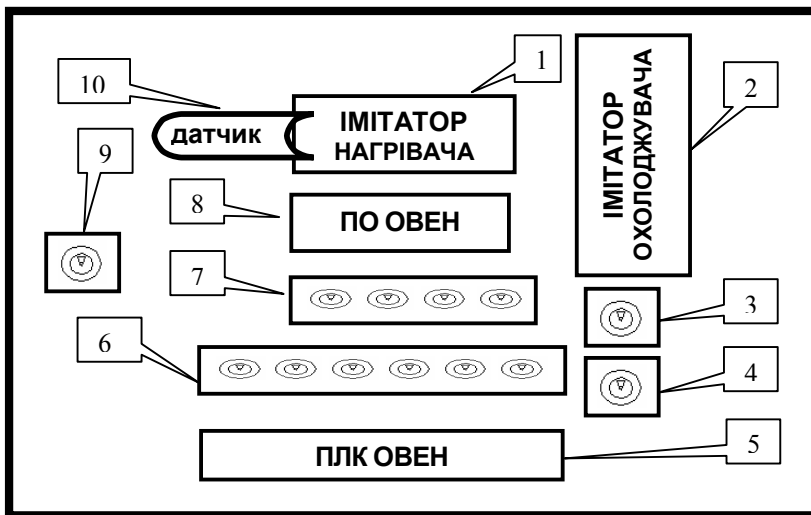


Рисунок 1.1 – Загальний вигляд стенда

На рис. 1.2 наведено принципову електричну схему стенда, а в табл. 1.1 – специфікацію стенду. Структурна схема комунікаційних зв'язків стенда зображена на рис. 1.3.

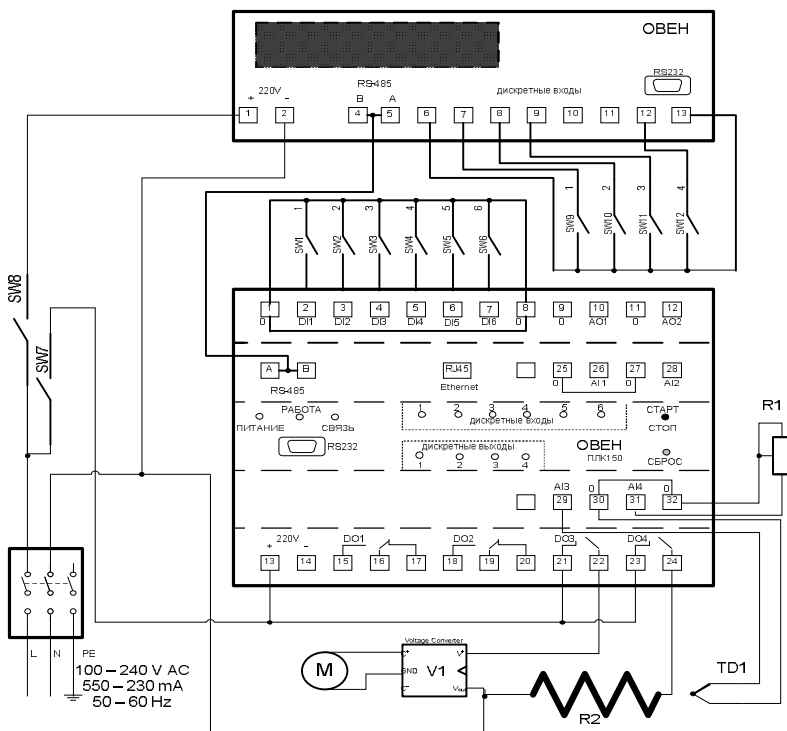


Рисунок 1.2 – Принципова схема стенда

Принцип роботи стенда такий. Клавiші **SW7**, **SW8** призначені для включення живлення ПЛК та ПО відповідно. За допомогою перемикачів **SW1...SW6** дискретні сигнали подаються на входи **DI1...6** ПЛК, а за допомогою перемикачів **SW9...SW12** – на входи **Bx.1**, **Bx.2**, **Bx.3** та **Bx.6** ПО. Вони можуть імітувати сигнали дискретних датчиків («ВМК.»/«ВИМК.»), управляти режимами роботи («Руч.»/«Автомат.») або технологічним процесом («Більше.»/«Менше»). Імітатори нагрівача та охолоджувача підключені до дискретних виходів, причому нагрівач

підключений безпосередньо до релейного виходу *DO4* ПЛК, а охолоджувач, через перетворювач напруги 220/9 В, – до виходу *DO3* ПЛК. Змінний опір підключений до аналогового входу *A1/4* ПЛК. Термопара *ТХК1479* підключена до аналогового входу *A1/3* ПЛК. Під час проведення лабораторних робіт панель оператора та дискретні датчики, що до неї приєднанні, не використовуються.

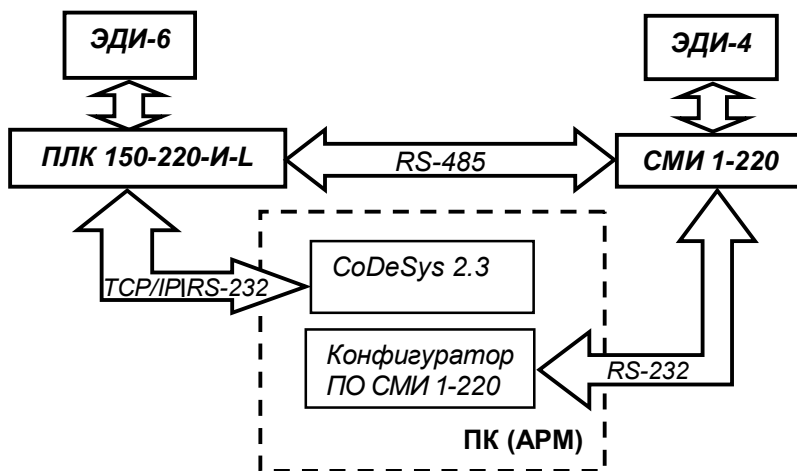


Рисунок 1.3 – Структурна схема стенда

Таблиця 1.1 – Специфікація елементів стенда

Перетворювач напруги <i>V1</i>	<i>БП 220/9В пост.</i>
Змінний опір <i>R1</i>	<i>0...1 кОм</i>
Нагрівальний резистор <i>R2</i>	<i>ПТЭВ-100, 750 Ом</i>
Вентилятор обдування <i>M</i>	<i>DC FAN 12V/0.1A</i>
Датчик температури <i>TD1</i>	<i>ТХК1479</i>
Перемикачі <i>SW1...SW6, SW9...SW12</i>	<i>MTS-1</i>
Клавіші вмикання <i>SW7, SW8</i>	<i>220В/5А</i>

1.3. Порядок виконання роботи

Виконання лабораторної роботи складається з наступних етапів:

1) Створення проекту в *CoDeSys V2.3*:

- вибір цільового ПЛК (*target*-файла) та мови програмування головного програмного організаційного компоненту (*POU*) – *PLC_PRG*, привласнення імені проекту;
- конфігурування ПЛК у вкладці *Ресурси* проекту;
- підключення додаткових бібліотек (при необхідності);
- написання програми користувача;
- запам'ятовування проекту та його компілювання;
- створення візуалізації для відображення змінних проекту.

2) Підключення ПК до ПЛК та завантаження до нього машинного коду проекту.

3) Налаштування проекту в покроковому режимі та перевірка роботи програми користувача для управління технологічним процесом.

1.4. Хід виконання роботи

1.4.1. Створення проекту в CoDeSys V2.3

Завантажте програмне середовище *CoDeSys 2.3*. Для цього ви-



користайте ярлик на робочому столі або запустіть файл *Codesys.exe* в директорії *C:\Program Files\3S Software\CoDeSys V2.3*.**. Також програмне середовище можна запустити, використовуючи кнопку *ПУСК* і послідовно вибираючи потрібну команду в меню, що спливає.

Якщо в загальних властивостях середовища не активовано флаг для завантаження останнього проекту, то відкриється пусте вікно. Для подальших дій необхідно, щоб були встановленні *target*-файли виробника контролерів (тобто опис структури ПЛК) в середовищі *CoDeSys 2.3*.

Взагалі, після встановлення комплексу *CoDeSys 2.3*, потрібно самостійно завантажити потрібні *target*-файли. Ця процедура здійснюється за допомогою програмного модуля *InstallTarget.exe*, який знаходиться в тій директорії, що і файл для запуску середовища *CoDeSys 2.3*. Далі потрібно діяти таким чином. По-перше, за допомогою кнопки «...» в верхній частині вікна вибрати потрібну директорію з *target*-файлами для встановлення та

натиснути на кнопку *OK*. По-друге, використовуючи кнопку *Open...*, знайти каталог з потрібним *target*-файлом. В результаті у вікні ліворуч *Possible Targets* з'явиться потрібна директорія (*PLC150.I_L*) з файлом *plc.trg*. Потретьє, натиснути на кнопку *Install* і далі – на кнопку *Close*. На рис. 1.4 зображено вікно програми-установника *InstallTarget* з потрібним *target*-файлом перед його завантаженням.

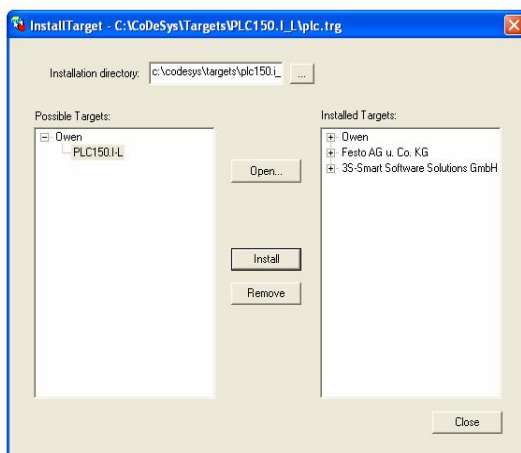


Рисунок 1.4 – Вікно з потрібним *target*-файлом перед його встановленням в *CoDeSys 2.3*

Отже, зараз все готово для подальшого створення проекту. Тому в меню *Файл* виберіть команду *Создать* або натисніть мишею на відповідну кнопку в панелі інструментів. В наступному вікні *Настройка целевой платформы*, використовуючи кнопку «▼» праворуч від поля *Конфигурация*, виберіть в випадаючому вікні потрібну модель контролера (*PLC 150-IL*). Відкриється вікно з вкладками для налаштувань параметрів ПЛК. Залишіть без змін параметри ПЛК. Підтвердіть свій вибір цільового контролера натисненням на кнопку *OK*. Далі, у вікні *Новый программный компонент* виберіть тип *POU* – програму, а також мову реалізації – *CFC* (графічна мова, яка використовує логічні елементи). Після всіх проведених операцій відкриється робоче вікно середовища програмування. Зовнішній вигляд вікна програмного середовища показано на рис. 1.5.

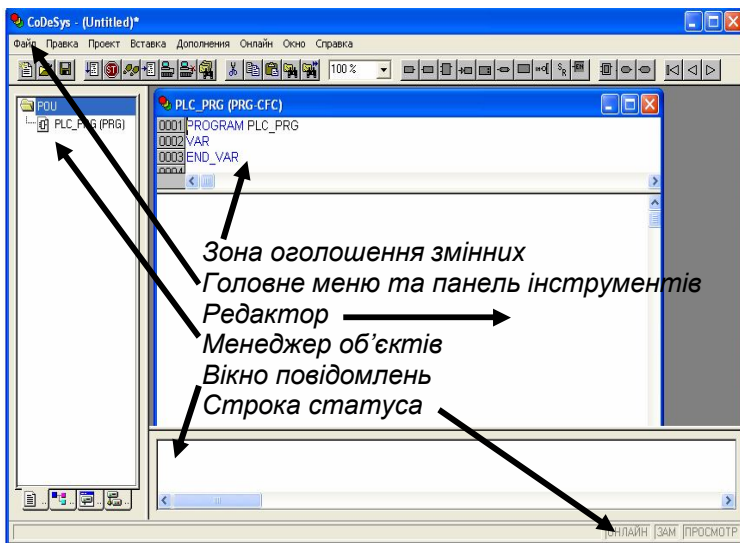


Рисунок 1.5 – Вікно з відкритим проектом

Використовуючи стандартні команди *Сохранить* або *Сохранить как...* в меню *Файл* або відповідну кнопку на панелі інструментів, збережіть проект на жорсткому диску ПК. Ім'я проекту має бути таким: *lr_1_name.pro*.

1.4.2 Конфігурування ПЛК

Далі необхідно провести конфігурування ресурсів фіксованого набору модулів ПЛК (дискретних та аналогових входів/виходів). Всі змінні, які оголошені в ресурсах ПЛК, є глобальними, причому обмін даними з програмою користувача здійснюється через спеціальну область пам'яті контролера – *пам'ять вводу-виводу*.

ПРИМІТКА. Розмір пам'яті вводу-виводу може бути обмежений ліцензією *CoDeSys*, що встановлена на контролер *ПЛК1xx*. Ліцензійне обмеження розповсюджується на контролери, останньою буквою в назві яких стоїть латинська «L». Тоді пам'ять вводу-виводу обмежена розміром 360 байт, при цьому 122 байти відводяться для пам'яті введення (%I), 234 байти – під пам'ять виведення (%Q), а 4 байти, що за-

лишилися – під спеціальну пам’ять (%M). Для контролерів, останньою буквою в назві яких стоїть латинська «M», обмежень у розмірі пам’яті вводу-виводу немає. За умовчанням сумарний обсяг пам’яті вводу (%I) і виводу (%Q) встановлений таким, що дорівнює **16Кб**, а це достатньо для більшості завдань.

Отже, в середовищі *CoDeSys 2.3* був створений проект з ім’ям *lr_1_name.pro* для цільової платформи *ОВЕН150-І.І* та з мовою програмування *CFC*. Далі перейдіть до вкладки *Ресурси* та виберіть утиліту *Конфігурація ПЛК*. Взагалі для конфігурування ресурсів ПЛК надається докладна інструкція [4].

По-перше, встановіть мінімальний час циклу ПЛК. Для цього у вкладці *Параметри модуля* клікніть по місцю пересічення рядка *Mincyclelength* та стовпчика *Значение*, а в рамці з пунктиру, що з’явиться, встановіть значення часу циклу – **10 мс**. Далі кліком миші по знаку «+», який знаходиться ліворуч поряд зі слотом *PLC150.I-L*, відкрийте його вміст. З’явиться дерево з фіксованим набором ресурсів ПЛК, екранна форма якого зображена на рис. 1.6.

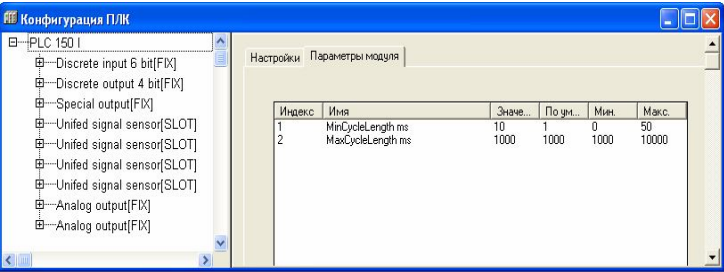


Рисунок 1.6 – Вікно з деревом ресурсів ПЛК

По-друге, проведіть конфігурування каналів дискретних входів/виходів та аналогових входів контролера. Для цього, аналогічно попередньому пункту, кликнувши по значку «+», відкрийте слоти ресурсів ПЛК: *Discrete input 6 bit (FIX)* – з каналами дискретних входів та *Discrete output 6 bit (FIX)* – з каналами дискретних виходів. Привласнення символічного імені каналу (входу або виходу) проводиться таким чином: подвійним кліком лівої кнопки миші натисніть на букви «АТ» у відповідно-

му рядку. Відкриється пуста рамка з пунктиром для редагування, до якої введіть ім'я змінної, використовуючи латинські літери, знаки та цифри. Причому, починати символічне ім'я з цифри заборонено.

Нехай для управління установкою буде використаний перемикач типу «сухий контакт», який підключений до одного з дискретних входів ПЛК за допомогою емулятора сигналів *ЭДИ-6*. Надайте першому дискретному входу ім'я *in1*. Подібним чином привласніть ім'я п'яти

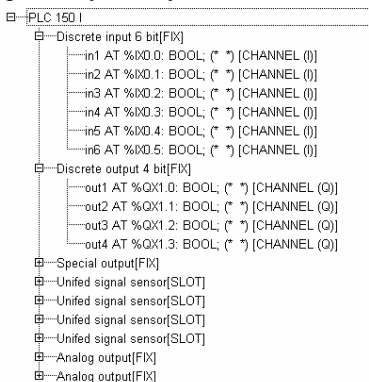


Рисунок 1.7 – Вікно з відкритими слотами вхідних та вихідних каналів ПЛК

входам, що залишилися. Далі налаштуйте чотири вихідні канали з підключеними до них вихідними пристроями, якими буде управляти ПЛК згідно з програмою користувача. Для цього привласніть символічні імена каналам, наприклад *out1*, *out2*, *out3* та *out4*. В результаті буде отримана наступна конфігурація дискретних вхідних та вихідних каналів (див. рис. 1.7).

Для конфігурування аналогових входів ПЛК в його ресурсах є чотири окремих слоти *Unifed signal sensor [SLOT]*. Кожний слот відповідає одному фізичному аналоговому входу ПЛК та вміщує два канали: власне вимірювальний (типу *REAL* з коментарем *Value*) та канал визначення часу циклу вимірювання (типу *WORD* з коментарем *Circular time*). Перший канал потрібен для зберігання поточного значення параметра, а другий – для вирахування сигналу регулювання під час реалізації ПД-регулювання. Слоти аналогових вхідних каналів можуть бути налаштованими на оброблення сигналів від джерел сигналу різного типу: уніфікованого датчика сигналу (встановлений за умовчанням), термометра опору, термоелектричного перетворювача (термопари) та контактного датчика («сухий контакт»). Замінити тип джерела можна за допомогою контекстного меню. Для цього лівою кнопкою миші слід помітити потрібний слот

датчика. В результаті він буде відмічений рамкою з точок. Далі натисніть праву кнопку миші та у вікні, що випадає, виберіть потрібне джерело сигналу, як це показано на рис. 1.8. На цьому рисунку показано порядок заміни уніфікованого датчика сигналу на термопару.

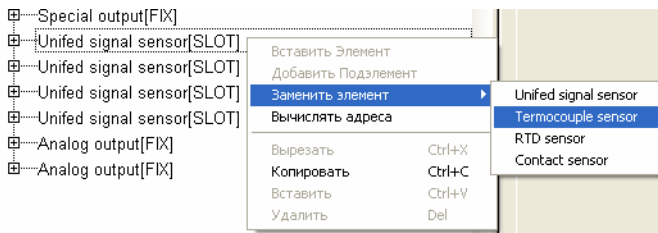


Рисунок 1.8 – Порядок заміни джерела сигналу в слоті ПЛК

Для кожного типу джерела в першому рядку параметрів – *Параметри модуля* – можна вказати конкретний тип первинного перетворювача сигналу. Для уніфікованого датчика сигналу це може бути уніфікований струмовий сигнал (за умовчанням, $0...20\text{ mA}$), сигнал напруги або резистивний сигнал. Якщо в якості джерела був вибраний термометр опору, то в переліку датчиків доступні платинові (за умовчанням встановлений тип *ТСП-50*, з НСХ, у якої коефіцієнт дорівнює 1.385), мідні та інші перетворювачі з різними НСХ. Для налаштування на приймання сигналу від термопар також можливий вибір їхнього типу (за умовчанням встановлений тип термопар – *ТХК (ТПЛ)*). У роботі [3] наведено всі типи джерел сигналів з їхніми позначеннями в конфігурації параметрів ресурсів ПЛК. Якщо потрібно збільшити кількість дискретних входів в ПЛК, то це можна зробити за рахунок аналогових входів. Для цього у відповідному слоті замініть аналоговий перетворювач на контактний датчик типу «сухий контакт» – *Contact sensor* (див. рис. 1.8). Отже, розглянуто основні правила роботи з дискретними входами/виходами та аналоговими входами контролера. Додатково в ПЛК є можливість формування ШІМ-сигналів на дискретних виходах ПЛК. Докладно це буде розглянуто в окремій лабораторній роботі. Але ознайомитися з порядком налаштування дискретного виходу на режим роботи ШІМ-регулювання можна в роботах [3, 4] та в ЛР №3.

Розглянемо далі доступні слоти фіксованого набору ресурсів ПЛК (див. рис.1.6). Це спеціальний (*Special output (FIX)*) та аналогові виходи (*Analog output (FIX)*). Модуль спеціального дискретного виходу містить бітову змінну, що управляє спеціальним устаткуванням (пристроєм подавання звукового сигналу або світлодіодом на верхній панелі). При значенні змінної *TRUE* вмикається спеціальне устаткування. Аналогові виходи відносяться до стаціонарних модулів. Залежно від типу встановлених в ПЛК ЦАП існують три варіанти налаштування модуля аналогових виходів:

- універсальний, який програмно настроюється або на ЦАП «параметр-струм», або на ЦАП «параметр-напруга»;
- фіксований, призначений для ЦАП «параметр-струм»;
- фіксований, призначений для ЦАП «параметр-напруга».

В лабораторних стендах встановлено *ПЛК150.И-Л* з виходами типу «струм 4...20 мА», які неможливо замінити на інші. Тобто в розпорядженні розробника СПЗ є лише струмовий вихід.

Отже, налаштуйте аналогові входи на оброблення сигналів від перетворювачів, які встановлені на стенді згідно із принциповою схемою (див. рис. 1.2). Так, до третього аналогового входу підключена термопара типу *ТХК*, а до четвертого – змінний резистор номіналом *0...1 кОм*. Зробіть налаштування вхідних каналів відповідно до екранних форм, що зображені на рис. 1.9 та рис.1.10.

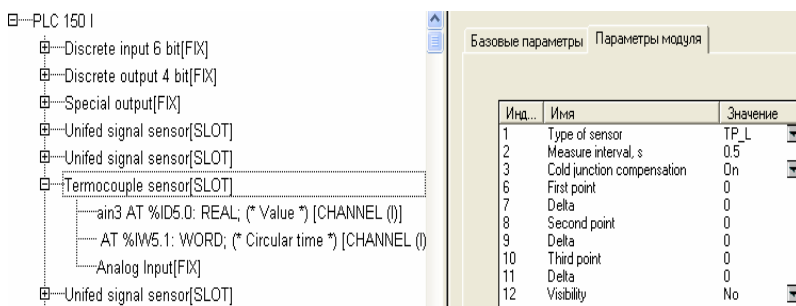


Рисунок 1.9 – Вікно налаштування аналогового входу для оброблення сигналу від термопари

Зверніть увагу на налаштування аналогового входу для оброблення сигналу від резистивного датчика. В рядку налаштування верхньої межі вкажіть число **5000**. Це потрібно для погодження діапазону зміни реального резистора ($0...1\text{ кОм}$) з типом програмного перетворювача – **RO_5000** в першому рядку параметрів модуля. В такому випадку значення змінної **ain4** буде збігатися з реальним значенням резистора, тобто $0...1000\text{ Ом}$. Якщо вказати інше значення верхньої межі, наприклад, на рівні **1000**, то значення змінної буде знаходитись в діапазоні $0...200$ одиниць. Це налаштування має назву «масштабування» параметра і може бути використане для перетворення параметра в відсоткове значення.

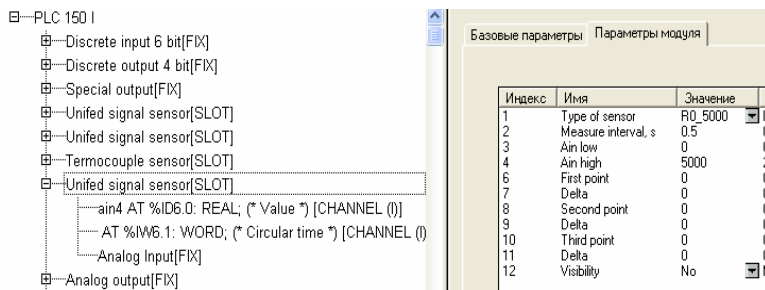


Рисунок 1.10 – Вікно налаштування аналогового входу для оброблення сигналу від змінного резистора

Насамкінець, налаштуйте аналоговий вихід ПЛК **АО1**, надавши йому символічне ім'я **aout1**. Цей вихід потрібен для формування управляючого сигналу для виконавчого механізму, наприклад, для пневматичного регулятора (**МІМ**), який через електропневмоперетворювач (**ЕПП**) з'єднаний з ПЛК. В якості програмного задатчика сигналу управління будемо використовувати змінний резистор, який підключений до входу **А/4**. Але для цього потрібно розробити відповідну програму перетворення поточного значення резистора в уніфікований струмовий сигнал **4...20mA**. Додатково розробіть логічну програму управління дискретними виходами ПЛК залежно від стану дискретних датчиків на його входах.

1.4.3. Розробка програми користувача

Поверніться з вікна налаштувань ресурсів ПЛК до вікна редактора програмних організаційних компонентів. Для цього в менеджері проекту, що знаходиться ліворуч (див. рис.1.5), клікніть мишею по вкладці *POU*, а далі – по головному *POU PLC_PRG* – для його відкриття.

Логічна програма для управління дискретними входами повинна працювати таким чином: виходячи з комбінацій логічних умов, потрібно вмикати або вимикати вихідні прилади, які підключені до ПЛК. Також необхідно вимірювати температуру та опір змінного резистора. Причому в програмі необхідно перетворювати поточне значення резистора в уніфікований струмовий сигнал *4...20mA*. При необхідності, поточне значення температури можна таким же чином перетворити в уніфікований струмовий сигнал.

Отже, програма користувача буде виглядати так, як це показано на екранній формі (див. рис. 1.11).

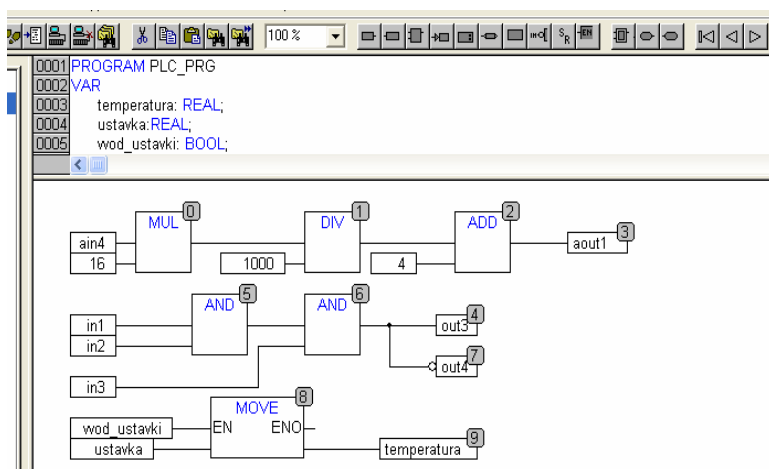


Рисунок 1.11 – Вікно програми користувача

Після створення програми користувача знову збережіть проект. Далі проведіть його компілювання, вибравши команду *Компілювати* або *Компілювати все* в меню *Проект* або натиснувши на функціональну кнопку «F11». Функціональна кнопка відповідає лише першій ко-

Онлайн	Окно	Справка
Подключение		Alt+F8
Отключение		Ctrl+F8
Загрузка		
Старт		F5
Стоп		Shift+F8
Сброс		
Сброс (холодный)		
Сброс (заводской)		
Переключить точку останова	F9	
Диалог точек останова		
Шаг поверху	F10	
Шаг детальный	F8	
Один цикл	Ctrl+F5	
Записать значения	Ctrl+F7	
Фиксировать значения	F7	
Освободить фиксацию	Shift+F7	
Диалог Запись/Фиксация	Ctrl+Shift+F7	
Показать стек вызовов...		
Отображать поток выполнения		
✓ Режим эмуляции		
Параметры связи...		
Загрузка исходных текстов		
Создание загрузочного проекта		
Записать файл в ПЛК		
Читать файл из ПЛК		

Рисунок 1.12 – Вікно з командами меню *Онлайн*

мандою *Старт*, або натиснувши на функціональну кнопку «F5». Далі в меню *Онлайн* стануть доступними команди для налагодження проекту, котрі зображені на рис. 1.12. Зауважимо, що в рядку статусу, який знаходиться внизу вікна (див. рис. 1.5), буде відображатися стан середовища програмування – *ОНЛАЙН*, режим – *ЭМУЛ.*, та статус емулятора – *ЗАПУЩЕНО*. Відключитися від емулятора слід за допомогою команди *Отключение* (див. рис. 1.12) в меню *Онлайн*.

1.4.4. Підключення до ПЛК та завантаження до нього коду проекту

Завантаження готових проектів до ПЛК можливе за допомогою двох інтерфейсів: послідовного – *RS-232* та локальної мережі *Ethernet* з протоколом *TCP/IP*. Отже, після компілювання програми, з'єднайтесь з

манді. Різниця між командами полягає в тому, що перша команда виконується тільки для активного вікна з *POU*. Виконання другої команди призводить до компілювання всього проекту. Якщо в програмі немає помилок, то компілювання пройде успішно, і проект буде готовий до завантаження до ПЛК.

Далі можна діяти двома шляхами: або налагодити проект в режимі емуляції ПЛК або завантажити проект до реального ПЛК. У першому випадку достатньо в меню *Онлайн* поставити прапорець поряд з рядком *Режим эмуляции*, а потім вибрати команду *Подключение*. Після завантаження проекту до емулятора потрібно його запустити ко-

ПЛК за допомогою програмного комунікаційного модуля та завантажить код проекту до ПЛК. Для цього потрібно в меню *Онлайн* зняти прапорець поряд з рядком *Режим емуляції*. Далі потрібно активувати вікно налаштування параметрів програмного комунікаційного модуля. Для цього натисніть на команду *Параметри зв'язи...* (див. рис. 1.12). Відкриється вікно, екранна форма якого зображена на рис. 1.13. Натисніть на кнопку *New...* для відкриття вікна налаштування нового каналу *Communication Parameters: New Channel* з переліком можливих драйверів каналів зв'язку з ПЛК. Розглянемо два варіанти підключення: за послідовним інтерфейсом *RS-232* та через *Ethernet*.

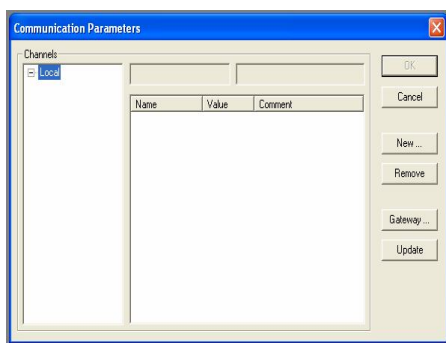


Рисунок 1.13 – Вікно налаштування модуля *Communication Parameters*

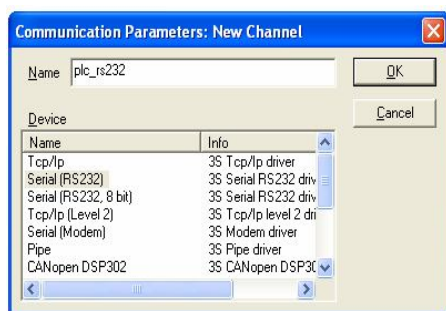


Рисунок 1.14 – Вікно вибору драйвера для каналу зв'язку з ПЛК

Для з'єднання за послідовним інтерфейсом для зв'язку через порт *Debug RS-232* ПЛК в випадаючому переліку доступних каналів виберіть другий рядок – *Serial (RS232)*. У полі *Name* введіть ім'я каналу, наприклад, *plc_rs232*. У результаті буде отримане вікно, яке зображене на рис. 1.14. Натисніть на кнопку *OK*, далі відкриється наступне вікно для налаштування параметрів *COM*-порту. Введіть потрібні дані в поля рядків з назвами параметрів стовпчика *Value*. Для зміни параметра слід двічі клікнути лівою кнопкою миші по наявному значенню параметра і, перегортаючи список доступних значень стрілками на клавіатурі ПК, вибрати нове зна-

чення. Для вводу нового значення натисніть кнопку *Enter* на клавіатурі. Вкажіть номер *COM*-порту, до якого фізично підключено ПЛК, швидкість передавання даних та формат кадру. Налаштуйте канал відповідно до екранної форми, що показана на рис.1.15. Завершіть налаштування *COM*-порту натисненням на кнопку *OK*.

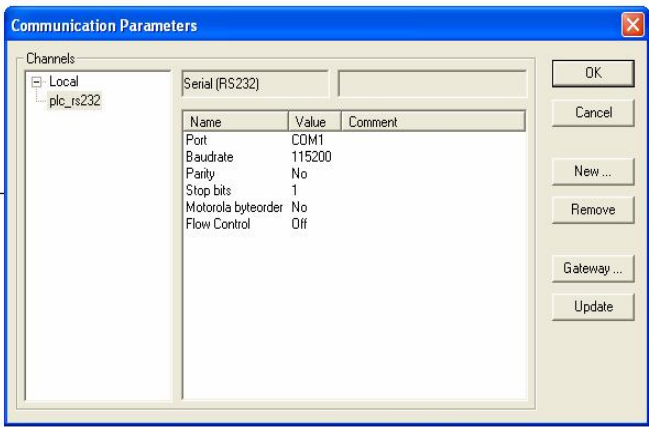


Рисунок 1.15 – Вікно налаштувань параметрів *COM*-порту

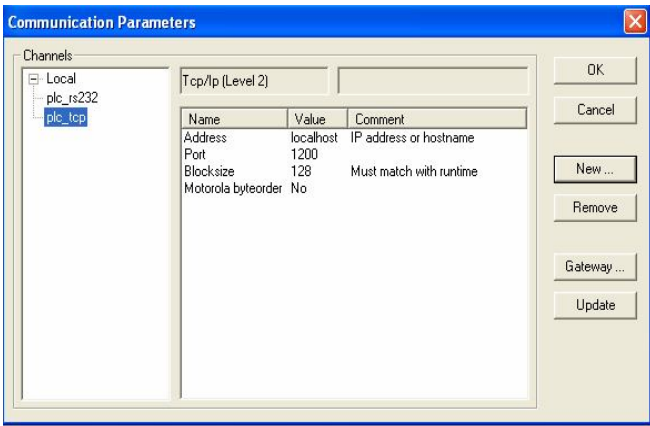


Рисунок 1.16 – Вікно налаштувань параметрів *Ethernet*-каналу

Для налаштування зв'язку з ПЛК по *Ethernet*-каналу потрібно в вікні *Communication Parameters: New Channel* (див. рис. 1.14) вибрати четвертий рядок з драйвером *Tcp/Ip (Level 2)*. Далі дійте, відповідно до попередніх інструкцій, до моменту, коли з'явиться вікно налаштування параметрів *Ethernet*-каналу. Його вигляд буде подібним до екранної форми, що зображена на рис.1.16. Відмітимо, що налаштування зв'язку з використанням *Ethernet*-каналу за протоколом *TCP/IP* має деякі особливості. Це стосується заповнення першого рядка з ім'ям *Address*, де вказується *IP*-адреса ПЛК замість значення *localhost* в стовпчику *Value*.

Важливо, що усі ПЛК спочатку мають одну адресу – **10.0.6.10**, яка встановлюється під час виробництва на заводі. В процесі подальшого використання ПЛК можна змінити цю адресу на іншу, яка буде відповідати сегменту локальної мережі. Зміна *IP*-адреси контролера можлива за допомогою команди *SetIP 172.016.028.****, що встановлюється через *ПЛК-Браузер* в ресурсах контролера. При цьому попередньо має бути встановлений зв'язок з контролером через інтерфейс *Debug RS-232*. Крім того, можлива зміна *IP*-адреси за допомогою програми *EasyWorkPLC.exe*, яка знаходиться в директорії зі шляхом доступу *C:\Program Files\Owen\EasyWorkPLC*.**. Проте *MAC*-адреса ПЛК є оригінальною і не може бути зміненою. Її значення можна визначити на бирці, що наліплена збоку ПЛК праворуч.

Для налаштування підключення до ПЛК саме з адресою **10.0.6.10** потрібно, щоб контролер та комп'ютер знаходились в одній *IP*-підмережі. Якщо ПК, який знаходиться в локальній мережі, має додатково мережний адаптер, то налаштуйте його властивості, використовуючи послідовно кнопки *ПУСК* та *Сетевое окружение* і команду *Отобразить сетевое окружение* в оболонці *Windows*. Далі активуйте вікно налаштувань властивостей локального з'єднання *Протокол Интернет (TCP/IP)* та введіть адресу ПК (яка відрізняється від адреси ПЛК, але входить до його сегмента, наприклад **10.0.6.11**) та маску (**255.255.255.0**) у вікні *Дополнительно*. У випадку наявності лише одного адаптера, налаштування буде подібним до попереднього, але необхідно програмно відключитися від основного сегмента. Для збереження нового значення *IP*-адреси натисніть

на кнопку *Enter* на клавіатурі ПК і далі на кнопку *OK* у вікні налаштувань каналу зв'язку (див. рис. 1.16).

Після налаштування з'єднання знову виберіть команду *Подключение* в меню *Онлайн*, яка встановлює зв'язок з контролером. При цьому прапорець напроти рядка меню *Режим эмуляции* має бути знятим.

УВАГА! Для зміни інтерфейсу з'єднання необхідно провести перезавантаження ПЛК, натиснувши на «втоплену» кнопку *Сброс* на його передній панелі. Для цього використовуйте, наприклад, скрепку або викрутку.

1.4.5. Створення візуалізації проекту

У середовищі *CoDeSys 2.3* можливе створення візуалізації проекту. Для цього перейдіть у вікно редактора візуалізацій – *Визуализация* (третє за порядком вкладення менеджера проектів, див. рис. 1.5). Для цього в менеджері проектів клікніть мишею по вкладці *Визуализация*, а далі, використовуючи контекстне меню, виберіть команду *Добавить объект...* для створення нового об'єкта – *Новая визуализация*. Привласніть ім'я цій візуалізації, наприклад *vis1*, при цьому в *CoDeSys 2.3* дозволено використовувати лише латинські літери та цифри. З'явиться вікно для створення візуалізації з панеллю інструментів для створення графічних елементів та об'єктів. Далі за допомогою різних інструментів створіть графічні елементи з відповідними налаштуваннями їхніх властивостей.

У складі програми користувача, яка наведена на рис. 1.11, є глобальні дискретні та аналогові змінні. Це дискретні входи *in1*, *in2*, *in3*; виходи *out3*, *out4*; аналогові входи *ain3*, *ain4*; виход *aout1*. У програмі також присутні локальні змінні – *ustavka*, *vvod_ustavki* та *temperatura*. Різниця між глобальними та локальними змінними полягає в тому, що локальні змінні доступні в межах лише одного *POU*, а глобальні – в межах всього проекту.

Отже, у вікні візуалізації створіть графічні елементи для відображення значень та стану змінних. Спочатку створіть елемент *Эллипс* для відображення стану дискретного входу ПЛК. Проведіть налаштування властивостей створеного елемента. Для цього підведіть курсор миші до елемента та клікніть по ньому. У результаті він буде позначений контуром зі знаком «+» в центрі. Використовуючи контекстне меню, виберіть

команду *Конфигурировать*. Відкриється вікно, екранна форма якого зображена на рис. 1.17.

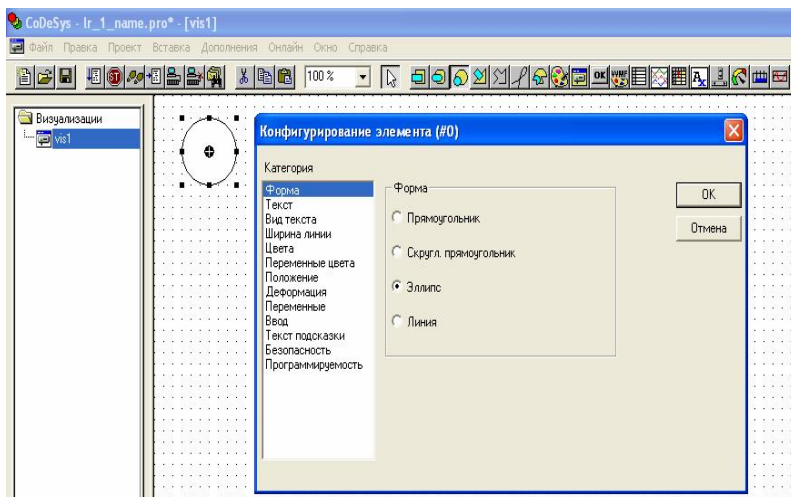


Рисунок 1.17 – Вікно налаштувань параметрів графічного елемента

Відкрийте категорію *Текст* та введіть в поле *Строка* наступний текст: **ВХОД 1**. У цій категорії можливе налаштування властивостей введеного тексту, які є стандартними для будь-яких текстових редакторів, наприклад *Microsoft Word*. Далі відкрийте категорію *Цвета*, в якій налаштуйте колір фону елемента залежно від значення змінної, яка буде зв'язана з ним. Наприклад, для нормального стану змінної (**FALSE**) натисніть на кнопку *Заливка* в полі *Цвета*, а у відкритому вікні виберіть жовтий колір та підтвердіть свій вибір натисненням на кнопку *OK*. Для аварійного стану змінної (**TRUE**) в полі *Тревожный цвет* також натисніть на кнопку *Заливка* та виберіть червоний колір. Далі необхідно вказати змінну, від значення якої буде залежати колір графічного елемента. Для цього відкрийте категорію *Переменные* і поставте курсор миші в третьому рядку з назвою *Изм. цвета:*. Далі введіть ім'я змінної в такому форматі – «*in1*» або використайте *Ассистент ввода*, який викликається функціональною кнопкою «F2». У відкритому вікні асистента клікніть по символу «+» напроти списку *Globale_Variables* і далі виберіть потрі-

бну змінну. Подібним чином створіть та налаштуйте інші графічні елементи для відображення стану дискретних входів та виходів.

Далі створіть графічні елементи *Прямоугольник* для відображення та власне для вводу значення змінної *ustavka*. Причому передавання в програмі користувача значення уставки змінної *temperature* буде здійснюватись, якщо змінна *vvod_ustavki* отримає значення *TRUE*. Таким чином створіть статичний елемент для відображення лише тексту, наприклад, *Введите значение уставки*. Під час введення тексту для переведення його на новий рядок необхідно одночасно натиснути на кнопки *Ctrl+Enter*. Налаштування динамічного тексту подібне до розглянутого вище для елемента *Эллипс*, але для відображення змінної в категорії *Текст* необхідно додати в другий рядок наступний запис *%3.1f град. С*. У категорії *Переменные* за допомогою *Ассистента ввода* введіть *PLC_PRG.ustavka* та додатково налаштуйте категорію *Ввод*. У цій категорії в зоні вводу поставте прапорець в рядок *Ввод в переменную 'Вывод_текста'*, у випадяючому меню за допомогою кнопки «▼» виберіть варіант *Цифр_панель* та вкажіть допустимий діапазон вводу змінної.

Для вводу значення змінної *vvod_ustavki* створіть графічний елемент типу кнопка «OK». У категорії *Ввод* поставте прапорець напроти поля *Переменная-кнопка*. У цій категорії за допомогою *Ассистента ввода* виберіть потрібну змінну – це буде *PLC_PRG.vvod_ustavki*.

ПРИМІТКА. Якщо з графічним елементом зв'язується локальна змінна, то перед нею обов'язково вказується програмний організаційний об'єкт, в якому вона створена. Це, наприклад, буде запис *PLC_PRG.vvod_ustavki*.

У цілому створіть візуалізацію згідно з прикладом, який наведено на рис.1.18.

Збережіть проект та завантажте його до ПЛК. Для запуску проекту натисніть на клавіатурі ПК функціональну кнопку «F5» або виберіть команду *Start* в меню *Онлайн*. За процесом можна спостерігати, якщо відкрити візуалізацію проекту або якщо стежити за значеннями змінних у вікні редактора програми.

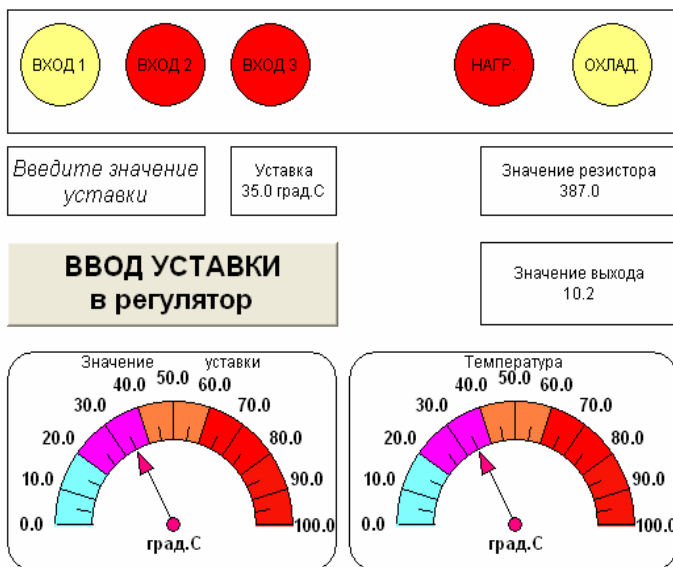


Рисунок 1.18 – Приклад візуалізації проекту

1.5. Перевірка роботи макета системи управління

1) Перевірте роботу програми користувача. Для цього вмикайте перемикачі, змінюйте опір резистору та вводіть значення локальних змінних і спостерігайте за всіма змінами на екрані монітора. Подайте проект викладачу для перевірки.

2) Складіть звіт в редакторі *Microsoft Office* відповідно до правил оформлення звітів: відомості про виконавця, назву та мету роботи, схему з'єднань, лістинг програми ПЛК, параметри налаштувань COM-порту для завантаження проектів.

1.6. Завдання для самостійної роботи

1) Розробіть програму користувача, в якій умовою вмикання та вимикання вихідних елементів будуть логічні вирази, що наведені в табл. 1.2. При цьому необхідно скласти таблицю істинності, тобто визначити необхідний стан входів для вмикання охолоджувача та нагрівача. Зробіть візуалізацію проекту.

Таблиця 1.2 – Варіанти умов для вмикання вихідних елементів

№ з/п згідно з журналом	Логічна умова (X1...X6 – дискретні входи)
1, 11, 21	$(X1 \text{ AND } X3) \text{ OR } (X4 \text{ AND } X5 \text{ AND NOT } X6)$
2, 12, 22	$(X2 \text{ OR } X3) \text{ AND } (NOT \text{ } X1 \text{ AND } X5 \text{ AND } X6)$
3, 13, 23	$(X1 \text{ AND } X3) \text{ OR } X4 \text{ AND NOT } X5 \text{ AND } X6)$
4, 14, 24	$(X2 \text{ AND } X3) \text{ XOR } (X1 \text{ AND } X5 \text{ AND } X6)$
5, 15, 25	$(NOT \text{ } X1 \text{ XOR } X3) \text{ AND } (X4 \text{ OR } X5 \text{ OR } X6)$
6, 16, 26	$(X2 \text{ AND NOT } X3) \text{ OR } (X4 \text{ AND } X5 \text{ AND } X6)$
7, 17, 27	$X2 \text{ AND } X3 \text{ AND NOT } X4 \text{ AND } (X1 \text{ XOR } X6)$
8, 18, 28	$(X1 \text{ OR } X3 \text{ OR } X4) \text{ AND } (NOT \text{ } X5 \text{ OR } X6)$
9, 19, 29	$(X2 \text{ XOR } X3) \text{ AND } (NOT \text{ } X4 \text{ OR } X1 \text{ OR } X6)$
10, 20, 30	$(X1 \text{ OR } X3 \text{ OR NOT } X4) \text{ AND } X5 \text{ AND } X6$

2) Розробіть програму користувача для дискретного управління об'єктом: нагрівачем або холодильником. У програмі необхідно реалізувати такі функції:

- вмикання сигналізації при виникненні будь-якої з аварій на об'єкті;

- вимикання об'єкта при виникненні будь-якої з аварій;

- вмикання об'єкта з кнопки, за умови відсутності аварій;

- вимикання об'єкта з кнопки.

Включення живлення об'єкта (*TEN* або *VENT*) або лампи сигналізації *LAMP* здійснюється залежно від результату логічної операції (РЛО) над чотирма вхідними дискретними змінними: *PUSK*, *STOP*, *AVAR* та *POGAR*. Якщо РЛО має статус «*TRUE*», то індикатор об'єкта буде синього кольору. У разі спрацювання датчиків індикатор *LAMP* буде червоного кольору, а індикатор об'єкта – сірого. Об'єкт включений до моменту натиснення на кнопку *STOP* або до спрацювання датчиків *AVAR* або *POGAR*. Нижче, в табл. 1.3 наведено варіанти входів та виходів ПЛК. Зробіть візуалізацію проекту згідно з прикладом, що зображений на рис. 1.19.

Таблиця 1.3 – Варіанти входів та виходів ПЛК

№ з/п згідно з журналом	PUSK	STOP	AVAR	POGAR	TEN або VENT	LAMP
1, 11, 21	DI1	DI2	DI3	DI4	DO3	DO2
2, 12, 22	DI2	DI3	DI4	DI5	DO4	DO1
3, 13, 23	DI3	DI4	DI5	DI6	DO3	DO2
4, 14, 24	DI4	DI5	DI6	DI1	DO4	DO1
5, 15, 25	DI5	DI6	DI1	DI2	DO3	DO2
6, 16, 26	DI6	DI1	DI2	DI3	DO4	DO1
7, 17, 27	DI1	DI2	DI3	DI4	DO3	DO2
8, 18, 28	DI2	DI3	DI4	DI5	DO4	DO1
9, 19, 29	DI3	DI4	DI5	DI6	DO3	DO2
10, 20, 30	DI4	DI5	DI6	DI1	DO4	DO1

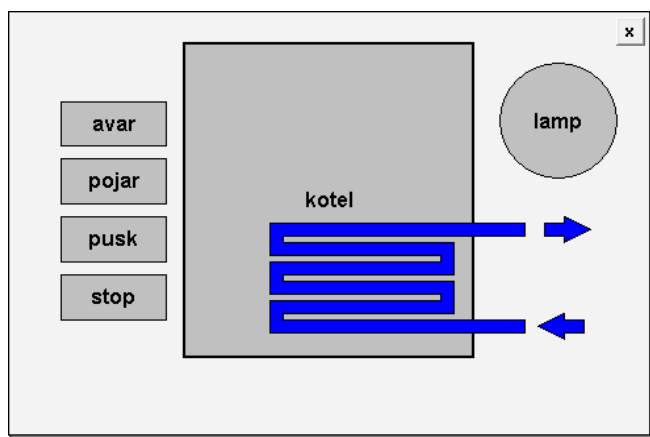


Рисунок 1.19 – Приклад візуалізації проекту для самостійної роботи

1.7. Контрольні питання

- 1) Наведіть коротку характеристику контролерів виробництва компанії *ОВЕН* моделей *ПЛК100\150\154*.
- 2) Який склад та можливості програмного комплексу *CoDeSys 2.3*?

- 3) Який склад проекту в середовищі *CoDeSys 2.3*?
- 4) Які інтерфейси використовують для завантаження проекту до ПЛК?
- 5) Яка послідовність створення проекту в *CoDeSys 2.3*?
- 6) Яка мета та порядок конфігурування ПЛК?

Лабораторна робота №2

СТРУКТУРУВАННЯ ПРОЕКТІВ ТА ОСНОВНІ ПРИНЦИПИ ВИКОРИСТАННЯ ТАЙМЕРІВ ТА ЛІЧИЛЬНИКІВ

2.1. Мета роботи

- закріплення теоретичних знань щодо використання програмних організаційних компонентів в проекті та їхнього структурування;
- засвоєння методики розробки програм користувача з використанням таймерів та лічильників.

2.2. Порядок виконання роботи

Виконання лабораторної роботи складатиметься з наступних етапів:

- 1) Створення проекту та розробка програми користувача з програмними організаційними компонентами – програмами (*PRG*) – для ознайомлення з роботою таймерів та лічильників.
- 2) Створення проекту та розробка програми користувача – управління насосом на трубопроводі подавання сировини.

2.3. Хід виконання роботи

2.3.1. Створення проекту в CoDeSys V2.3 для ознайомлення з роботою таймерів та лічильників

Проведіть підготовчі роботи відповідно до дій, що описані в ЛР№1 (див. пункт 1 в підр. 1.3 та пункти 1.4.1 – 1.4.3). Але при створенні головного *POU* в якості мови програмування виберіть мову *ST* (мова, що подібна до мови *Pascal*). У робочому вікні введіть рядки для виклику *POU* типу *PRG*, а саме для виклику програм з таймерами, лічильниками та з таймером реального часу. У результаті вигляд головного *POU* буде відповідати екранній формі, що зображена на рис. 2.1.

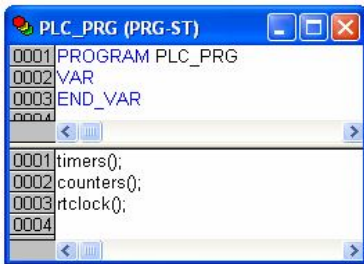


Рисунок 2.1 – Вікно головного *POU* проекту

умовчанням вони при створенні будуть типу *AND*. Виділіть кліком миші один із створених елементів і далі використайте асистент вводу для заміни елемента *ADD* на елемент *CTD* зі списку стандартних функціональних блоків, які входять до списку *Counter* бібліотеки *STANDARD.lib*. Вказана бібліотека автоматично підключається за умовчанням до проекту під час його створення. На рис. 2.2 зображена екранна форма, на якій показано спосіб вибору елемента декрементного лічильника *CTD (FB)* замість елемента *AND*, який створюється за допомогою *Ассистента ввода* (функціональна кнопка «F2») або способом *drag-and-drop*.

Далі в менеджері проектів за допомогою контекстного меню послідовно створіть три нових *POU* – програми з мовою реалізації *CFC*. Отже, зараз все готово для подальшого розроблення проекту.

Відкрийте для редагування один з створених *POU* – *PRG counters*. У робочому полі блока послідовно додайте три блочних елементи. За

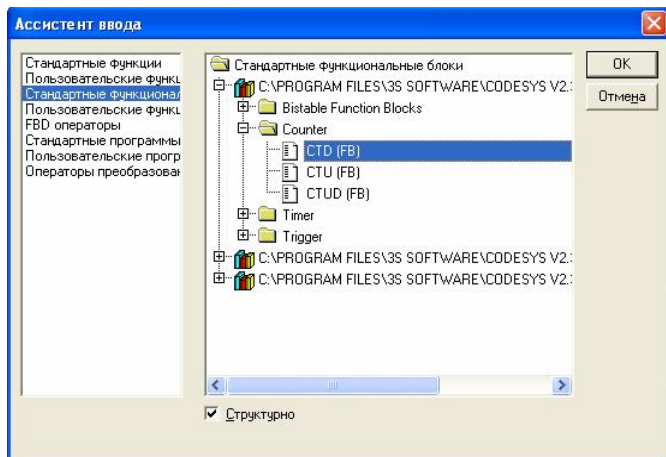


Рисунок 2.2 – Вікно вибору декрементного лічильника

Аналогічно замініть два елементи, що залишилися, на лічильники типу інкрементний (CTU) та інкрементно-декрементний (CTUD). Підключіть до входів та виходів лічильників відповідні вхідні та вихідні елементи.

Програма *counters* повинна виглядати так, як вона зображена на рис. 2.3. У полі оголошення змінних показані всі вхідні та вихідні елементи типу *BOOL* для управляючих входів та виходів і типу *WORD* – для відображення значення лічильників.

Аналогічним чином відредагуйте наступний *POU – PRG timers*. На рис. 2.4 показана екранна форма заповнення робочого вікна програми *timers*.

На рис.2.5 зображений приклад використання блока реального часу *RTC*, який входить до *POU* типу *PRG – rtclock*.

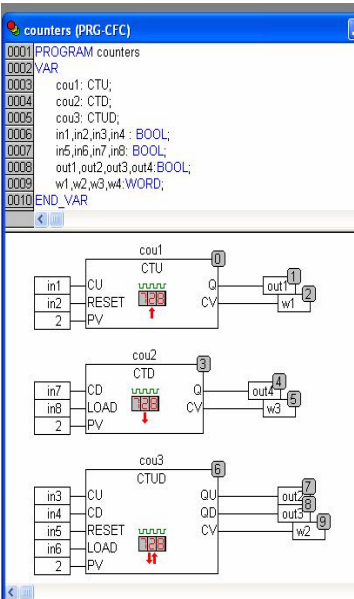


Рисунок 2.3 – Вікно програми *counters*

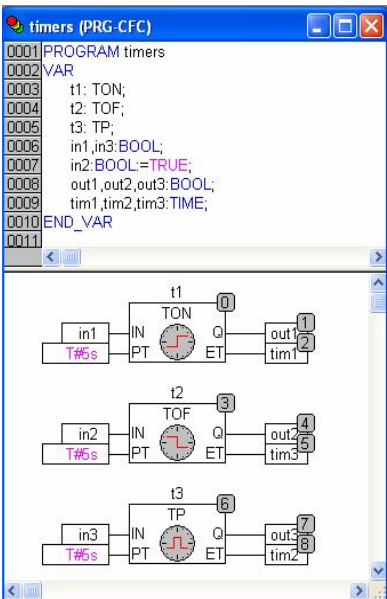


Рисунок 2.4 – Вікно програми *timers*

Збережіть проект під ім'ям *lr_2_1_name.pro*. Зробіть візуалізацію для одного з програмних компонентів за вказівкою викладача та подайте йому отримані результати.

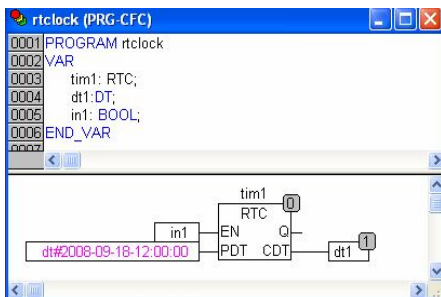


Рисунок 2.5 – Вікно програми *rtclock*

2.3.2. Створення проекту з програмою управління насосом

Розробіть проект з програмою управління насосом на трубопроводі подавання сировини з наступним алгоритмом. Кнопкою «ПУСК» вмикається електромотор насоса. Насос повинен працювати на

протязі 10 секунд. Далі він автоматично вимикається. Для вмикання насоса потрібно знову натиснути на кнопку «ПУСК». Одночасно в програмі потрібно рахувати кількість вмикань насоса. Проект розробляти з використанням програмного організаційного компонента – функціонального блока. Програма користувача буде відповідати екранній формі, що зображена на рис. 2.6. Зробіть візуалізацію процесу, подібну до зображеної на рис. 2.7. Далі збережіть проект під ім'ям *lr_2_2_name.pro*.

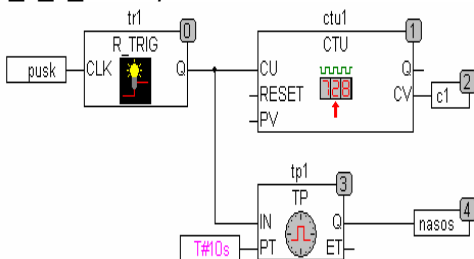


Рисунок 2.6 – Програма управління насосом

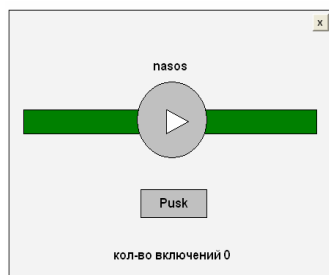


Рисунок 2.7 – Візуалізація програми управління насосом

Для запуску проекту з візуалізацією без запуску середовища *CoDeSys V2.3* створіть ярлик виклику проекту, як це робиться зазвичай в середовищі *Windows*. За допомогою контекстного меню відкрийте властивості ярлика. У вікні властивостей перейдіть на вкладку *Ярлик*, а в ній заповніть поля *Объект* та *Рабочая папка*. Для поля *Объект* рядок для заповнення буде таким: *"C:\Program Files\3S Software\CoDeSys V2.3*

CoDeSysHMI\CoDeSysHMI.exe"
/simulation"D:\OWEN\1_nas_vkl.pro".

У даному рядку перша частина вказує місце розташування файлу запуску програвача візуалізацій *CoDeSysHMI.exe*, а друга частина після символу «/» та параметра *simulation* указує місце розміщення проекту на жорсткому диску ПК. Параметр *simulation* вказує, що проект буде запущений в режимі симуляції. У поле *Рабочая папка* введіть шлях до директорії з проектом: *D:\OWEN*. Екранна форма налаштувань властивостей ярлика зображена на рис. 2.8. Взагалі порядок роботи з програвачем візуалізації можна отримати з довідкової допомоги, яка додається до середовища *CoDeSys V2.3* (див. розділ *CoDeSys HMI*).

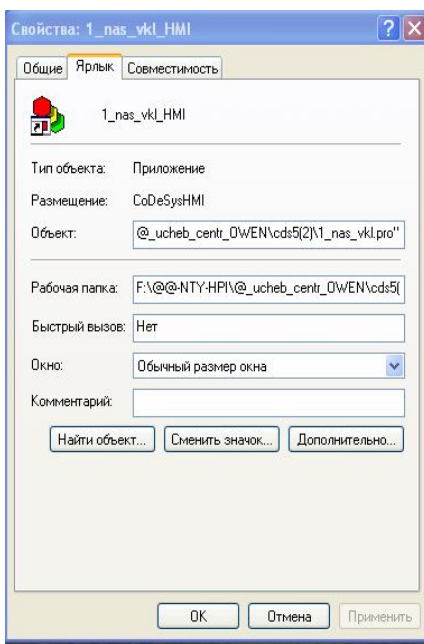


Рисунок 2.8 – Вікно налаштувань властивостей ярлика для виклику проекту

Подайте викладачу проект з можливістю його виклику через командну строку (ярлик) з реалізацією на емуляторі або на цільовій платформі.

2.3.3. Удосконалення проекту з програмою управління насосом

Розробіть проект з програмою управління насосом на трубопроводі подавання сировини з наступним алгоритмом: вимірювання часу роботи насоса та захист роботи насоса від «сухого ходу». Кнопкою «ПУСК» вмикається електромотор насоса. Нехай для забезпечення захисту насоса від «сухого ходу» в трубопроводі інтегрований датчик тиску. Якщо на протязі п'яти секунд не спрацює датчик тиску, то насос автоматично вимикається через три секунди. Доопрацюйте попередній проект з програмою управління таким чином, як це показано на рис. 2.9. Далі збе-

режіть проект під ім'ям *lr_2_3_name.pro*. Доопрацюйте попередню візуалізацію (див. рис. 2.7) таким чином, як це показано на рис. 2.10.

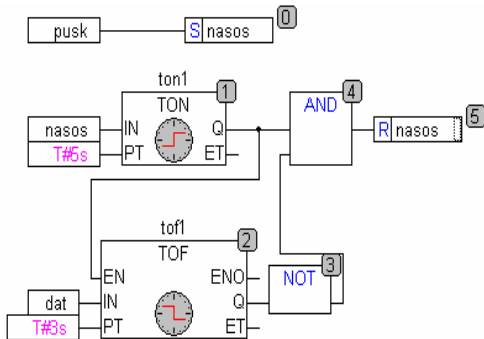


Рисунок 2.9 – Програма управління насосом з датчиком тиску

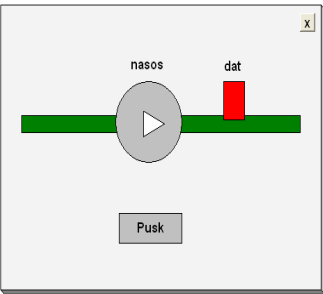


Рисунок 2.10 – Візуалізація проекту системи управління насосом

2.3.3. Створення бібліотеки користувача на прикладі проекту з функціональним блоком

Знову відкрийте проект з ім'ям *lr_2_2_name.pro*. Для створення власної бібліотеки з метою подальшого багатократного використання розробленого готового функціонального блока дійте таким чином.

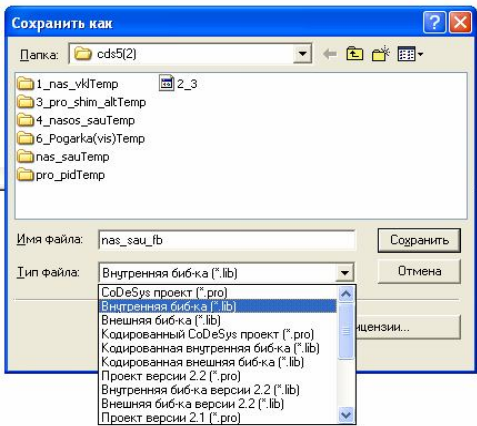


Рисунок 2.11 – Вікно для створення бібліотеки користувача

По-перше, збережіть проект з новим ім'ям.

По-друге, створіть *POU* функціональний блок, перенесіть до нього програму з головного *POU* (див. рис. 2.9), далі видаліть з проекту головний *POU* – *PLC_PRG*.

По-третє, в меню *Файл* виберіть команду *Сохранить как....* Далі у вікні, екранна форма якого зображена на рис.2.11, дайте на-

зву власному бібліотечному функціональному блоку у полі *Имя файла*: – *nas_sau_fb*. У полі *Тип файла*: виберіть в випадяючому меню тип бібліотеки – *Внутренняя биб-ка (*.lib)*. Далі натисніть на кнопку *Сохранить*. У результаті буде отримана бібліотека користувача з функціональним блоком для управління насосом.

2.4. Перевірка роботи макета системи управління

1) Перевірте правильність роботи проектів з розробленими програмами управління. Для цього вмикайте/вимикайте перемикачі та спостерігайте за всіма змінами у візуалізації на екрані монітора ПК. Подайте проекти викладачу для перевірки.

2) Складіть звіт в редакторві *Microsoft Office* відповідно до правил оформлення звітів: відомості про виконавця, назву та мету роботи, схему з'єднань, лістинги програм ПЛК, параметри налаштувань COM-порту для завантаження проектів.

2.5. Завдання для самостійної роботи

1) Розробіть проект, що реалізує дискретне управління насосом з використанням таймерів, а також датчика тиску аналогової дії.

Необхідно модифікувати проект *lr_2_3_name.pro* для випадку використання датчика тиску аналогової дії. Для імітування датчика тиску використовуйте змінний резистор, а поріг для відключення встановіть на рівні 10 % від всього діапазону (0...1000 одиниць).

Зробіть візуалізацію проекту, подібну до зображеної на рис. 2.10, але додайте графічний символ для відображення значення тиску в трубопроводі.

2) Розробіть проект, що реалізує дискретне управління насосом та електроклапаном. Як приклад використовуйте усі проекти, що запропоновані в даній роботі.

Необхідно додатково реалізувати у проекті наступні функції:

- якщо на протязі п'яти секунд датчик тиску досягне потрібного значення (30 % від всього діапазону), то вмикається випускний електроклапан;
- якщо тиск упаде до мінімального рівня (10 % від всього діапазону), то випускний електроклапан вимикається.

Зробіть візуалізацію проекту, подібну до зображеної на рис. 2.10, але додайте графічний символ для відображення стану електроклапана та значення тиску в трубопроводі.

2.6. Контрольні питання

- 1) Дайте коротку характеристику таймерів згідно з *IEC61131*.
- 2) Дайте коротку характеристику лічильників згідно з *IEC61131*.
- 3) Яким чином можна використовувати таймери та лічильники в програмах управління зовнішніми пристроями?
- 4) Дайте коротку характеристику бібліотеки *Standard.lib*.

Лабораторна робота №3

ПРИНЦИПИ РЕАЛІЗАЦІЇ 2-ПОЗИЦІЙНОГО ТА ШІМ-РЕГУЛЮВАННЯ

3.1. Мета роботи

- закріплення теоретичних знань щодо принципів дво-позиційного управління дискретними вихідними пристроями;
- закріплення теоретичних знань щодо принципів аналогового (ШІМ) управління дискретними вихідними пристроями;
- засвоєння порядку налаштування апаратного ШІМ-регулятора на прикладі конфігурування ресурсів ПЛК ОВЕН;
- розробка програм користувача для ШІМ-регулювання на прикладі використання стандартних функціональних блоків з бібліотеки *Util.lib*.

3.2. Порядок виконання роботи

Виконання лабораторної роботи складається з таких етапів:

- 1) Створення проекту та розробка програми 2-позиційного сигналізатора або регулятора параметра.
- 2) Створення проекту та розробка програми ШІМ-регулятора на прикладі використання імпульсного таймера.

3) Створення проекту та розробка програми ШІМ-регулятора на прикладі використання блока *BLINK* з бібліотеки *Util.lib*.

3.3. Хід виконання роботи

3.3.1. Створення проекту с програмою 2-позиційного сигналізатора параметра

Необхідно реалізувати у проекті такі функції:

- вимірювання температури за допомогою датчика, який підключений до аналогового входу;

- включення сигналізації (лампа), якщо температура буде вища від заданого першого значення (за умовчанням 150 °C);

- відключення сигналізації, якщо температура буде нижча від заданого другого значення, яке менше за перше (за умовчанням 100 °C).

Нехай аналоговий сигнал імітується змінним опором, який підключений до входу *A14* ПЛК (встановить верхню межу АЦП на рівні 1000

одиниць, тоді діапазон зміни параметра буде складати від 0 до 200 °C). Зробіть візуалізацію процесу, подібну до зображеної на рис. 3.1.

Збережіть проект з ім'ям *lr_3_1_name.pro*. Заповніть поле програми користувача згідно з лістингом:

```
PROGRAM PLC_PRG
  VAR
    ust1: REAL := 150;
    ust2: REAL := 100;
  END_VAR
  IF temp>ust1 THEN lamp:=TRUE;
  END_IF
  IF temp<ust2 THEN lamp:=FALSE;
  END_IF .
```

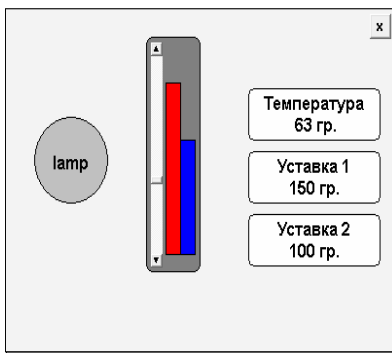


Рисунок 3.1 – Візуалізація проекту сигналізатора параметра

Подайте викладачу проект з можливістю його виклику через командну строку з реалізацією на емуляторі або на цільовій платформі.

3.3.2. Загальні відомості про принципи 2-х позиційного регулювання

РЕГУЛЯТОР – це пристрій, який призначений для підтримки контрольованої величини на заданому рівні. Для цього можна використовувати вихідний пристрій з двома станами: включено або виключено. Причому для підтримки заданого значення регульованого параметра можна використовувати різні типи виконавчих пристроїв, але всі вони можуть бути умовно розділені на дві групи: «нагрівачі» і «охолоджувачі».

«Нагрівачем» називають пристрій, включення якого повинне приводити до збільшення значення вимірюваного параметра.

«Охолоджувачем» називають пристрій, включення якого повинне приводити до зменшення значення вимірюваного параметра.

Алгоритм сигналізатора можна модифікувати, якщо додати до програми параметр гистерезис. Тобто замість двох уставок можна використати одну уставку з гистерезисом. Нижче наведено приклад програми терморегулятора в режимі прямого гистерезиса, тобто «нагрівача»:

```
PROGRAM PLC_PRG
VAR
    gist: REAL := 5;
    rele_b.0: BOOL;
END_VAR
IF (T_TXK<(ust-gist) ) THEN (*если температура меньше
уставки*)
    rele:=TRUE;              (*включить реле нагревателя*)
    rele_b.0:=TRUE;          (*выставит флаг включения реле*)
END_IF;
IF (T_TXK>(ust+gist) ) THEN (*если температура выше
уставки*)
    rele:=FALSE;             (*выключить реле нагревателя*)
    rele_b.0:=FALSE;         (*выставит флаг выключения реле*)
END_IF;
```

Аналогічним чином буде виглядати програма терморегулятора в режимі зворотного гистерезиса, тобто «охолоджувача». Але умовою вми-

кання вихідного пристрою буде $T_{TXK} > \text{ust} + \text{gist}$, а вимикання – $T_{TXK} < \text{ust} + \text{gist}$.

Створіть проект з ім'ям *lr_3_2_name.pro* згідно із завданням за номером в журналі та подайте проект викладачу.

3.3.3. Загальні відомості про принципи ШІМ-регулювання

Сигнал ШІМ – це імпульсний сигнал постійної частоти та змінної шпаруватості. За допомогою завдання шпаруватості (тривалість імпульсів) можна змінювати середню напругу на дискретному виході ПЛК.

Шпаруватість – одна з класифікаційних ознак імпульсних систем, яка визначає відношення періоду слідування сигналу до тривалості імпульсу. Величина, яка обернена шпаруватості, в англomовній літературі називається коефіцієнтом заповнення (від англ. *Duty cycle*).

Таким чином, для імпульсного сигналу справедливі такі співвідношення: $S = T / \tau = 1 / D$, де S – шпаруватість, D – коефіцієнт заповнення, T – період слідування імпульсів, τ – тривалість імпульсу.

Часте застосування в практиці знаходить сигнал з шпаруватістю, яка дорівнює двом, це сигнал типу «меандр».

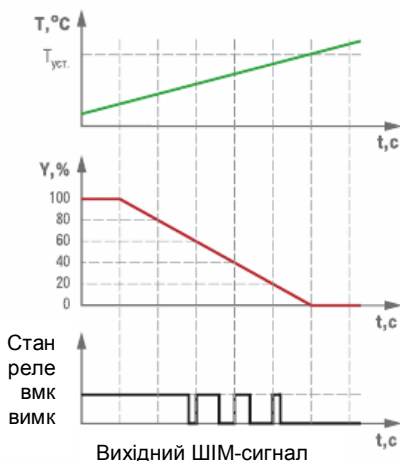
У режимі аналогового регулювання логічний пристрій розраховує відхилення E поточного значення контрольованої величини T від заданої уставки $T_{уст}$ (тобто, розузгодження). У результаті на виході регулятора виробляється аналоговий сигнал Y , який направлений на зменшення розузгодження E . Цей сигнал подається на виконавчий пристрій регулятора у вигляді послідовності імпульсів (ШІМ).

Якщо вихідний пристрій регулятора ключового типу (реле, транзисторна або симісторна оптопара, вихід для управління твердотільним реле), вихідний сигнал перетвориться в послідовність управляючих імпульсів з тривалістю D :

$$D = Y \frac{T_{\text{н\acute{e}}}}{100\%},$$

де D – тривалість імпульса, с; $T_{сл}$ – період слідування імпульсів, с (задається користувачем у програмі); Y – вихідний сигнал регулятора.

Принципи формування ШІМ-сигналу для «нагрівача» при різних значеннях вихідного сигналу Y показані на рис. 3.2. На рис. 3.3 показані діаграми ШІМ-сигналів для різних значень шпаруватості.



Вихідний ШІМ-сигнал
Рисунок 3.2 – Принцип формування ШІМ-сигнала

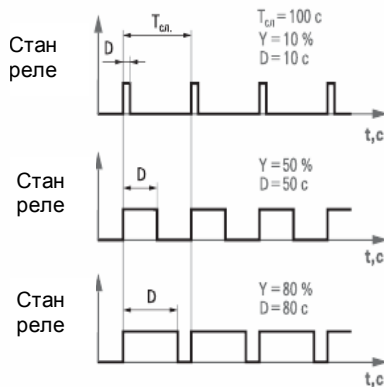


Рисунок 3.3 – ШІМ-сигнал для різних значень шпаруватості

3.3.4. Апаратний ШІМ-регулятор

Для закріплення теоретичних відомостей щодо принципів ШІМ-регулювання на прикладі апаратного дискретного виходу створіть новий проект з ім'ям *lr_3_3_name.pro*. В якості цільової платформи виберіть *ОВЕН ПЛК150.I-L*. Мову реалізації *POU PLC_PRG* оберіть *CFC*.

Для встановлення значення параметра шпаруватості знову використайте змінний резистор, який підключений до аналогового входу ПЛК. Для реалізації апаратного ШІМ-регулятора налаштуйте дискретний вихід ПЛК на роботу в режимі формування ШІМ-сигналу. Для цього перейдіть до вікна конфігурування ПЛК. Відкрийте фіксований слот дискретних виходів. Далі за допомогою миші викличте контекстне меню, а в ньому виберіть команду *Добавить Pulse-wide modulator*. Таким чином, отримаєте у складі фіксованого слоту дискретних виходів *Discrete output 4 bit [FIX]* додатковий канал *Pulse-wide modulator* типу *VAR*. На рис. 3.4 зображена екранна форма вікна зі створеним каналом ШІМ.

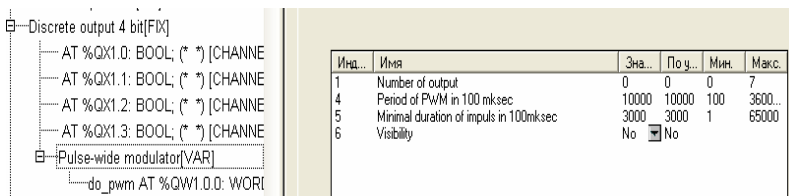


Рисунок 3.4 – Вікно налаштування каналу формування ШІМ-сигналу

У параметрах каналу необхідно вказати номер фізичного виходу, до якого підключений вихідний пристрій. Зауважимо, що нумерація виходів ПЛІК починається з нуля. Тобто якщо нагрівач підключений до виходу **DO4**, то в параметрах налаштувань модуля *Number of output* в першому рядку необхідно ввести число «три». Додатково в параметрах модуля необхідно встановити період слідування імпульсів необхідною кількістю часових інтервалів по **100 мкс**. За умовчанням встановлена кількість інтервалів дорівнює **10000**. Таким чином, період слідування імпульсів розраховується згідно з наступним виразом: $T=100\text{мкс} \cdot 10000=1\text{с}$. Максимальне можливий період становить **360000** інтервалів, що відповідає **360** секундам. Аналогічним чином можна встановити мінімальний час тривалості імпульсу. Це потрібно для забезпечення експлуатаційних характеристик вихідного пристрою.

Спочатку встановіть період слідування імпульсів на рівні **2 с**. Далі за допомогою змінного резистора можна встановлювати час тривалості імпульсу. Але при цьому потрібно врахувати, що значення резистора на аналоговому вході зберігається в змінній типу *REAL* і може набувати значень від **0** до **1000** (якщо в параметрах налаштувань встановити верхню межу на рівні **5000**). На виході каналу *Pulse-wide modulator* значення параметра зберігається в змінній типу *WORD* і може набувати значення від **0** до **65535**. Тому в програмі користувача значення резистора потрібно спочатку масштабувати (як це показано в ЛР№1, пункт 1.4.2), а потім перетворити в змінну типу *WORD*. Приклад програми зображений на рис. 3.5.

У результаті розроблений проект з можливістю ручного встановлення тривалості імпульсу при ШІМ-регулюванні. Але це лише мо-

дель, яка наявно показує, як працює ШІМ-регулятор. Тому зробить візуалізацію проекту, в якій створить тренди для відображення значення вхідного та вихідного параметрів на одному графіку. Подайте проект викладачу для перевірки. Побудуйте графік залежності тривалості імпульсу при ШІМ-регулюванні від значення параметра на вході ПЛК.

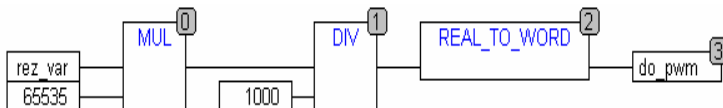


Рисунок 3.5 – Програма управління апаратним каналом ШІМ

3.3.5. ШІМ-регулятор на основі імпульсного таймера

Якщо в ПЛК не передбачена можливість апаратного формування ШІМ-сигналу (наприклад, немає модуля *Pulse-wide modulator*), то цей недолік можна усунути за рахунок програмного способу його формування. Але в цьому випадку потрібно одночасно формувати часові інтервали для періоду слідування імпульсів та власне управляти тривалістю імпульсів. Це завдання можна вирішити за допомогою стандартного блока *BLINK* з бібліотеки *Util.lib*.

Якщо при створенні проекту бібліотека не була підключена, зробить це зараз. Для цього за допомогою менеджера проектів перейдіть до його ресурсів. Подвійним кліком лівої кнопки миші по рядку *Менеджер библиотек* відкрийте вікно управління бібліотечними ресурсами. З'явиться вікно менеджера бібліотечних ресурсів, екранна форма якого показана на рис. 3.6. На цьому рисунку вікно розділене на декілька полів: поле ліворуч зверху для управління бібліотеками, поле ліворуч знизу зі структурою відміченої бібліотеки, поле праворуч зверху з змінними активованого елемента бібліотеки та поле праворуч знизу з позначенням вибраного елемента бібліотеки. У вікні для управління бібліотеками за допомогою правої кнопки миші активовано контекстне меню з відповідними командами: *Добавить библиотеку...*, *Удалить* та *Свойства...*. Зауважимо, що покажчик миші при цьому повинен знаходитися в полі з переліком раніше підключених бібліотек, тобто зверху ліворуч.

Отже, наступним кроком виберіть команду *Добавить библиотеку....* Далі з'явиться вікно *Открыть* з каталогом *Library*, який знаходиться в каталозі з програмним середовищем *C:\Program Files\3S Software\CoDeSys V2.3*.**. Відмітьте потрібну бібліотеку та натисніть на кнопку *Открыть*. Взагалі місце знаходження бібліотек можна встановити або змінити, якщо використати в меню *Проект* команду *Опции...*, а в вікні налаштування опцій вибрати категорію *Директории*.

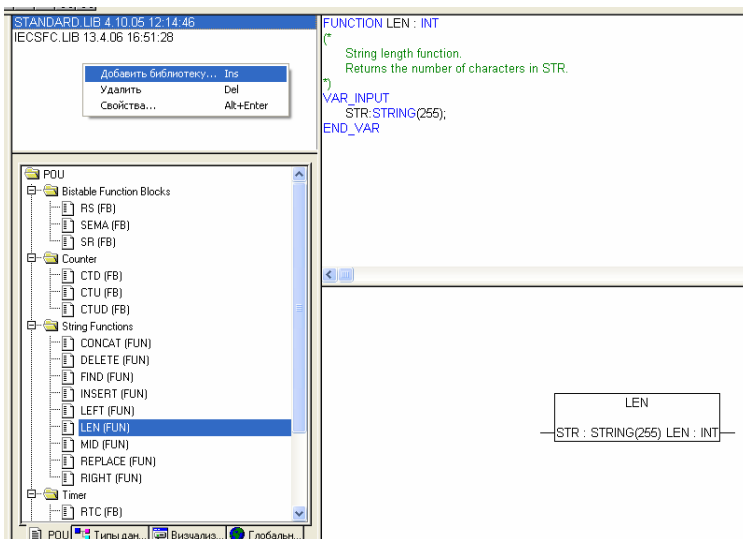


Рисунок 3.6 – Вікно менеджера бібліотечних ресурсів

Отже, після проведення підключення потрібної бібліотеки створіть проект з ім'ям *lr_3_4_name.pro*. В якості цільової платформи виберіть *OBEH ПЛК150.I-L*. Мову реалізації *POU PLC_PRG* оберіть *CFC*.

Як в попередніх прикладах, використовуйте змінний резистор для формування часових інтервалів. Вихідний канал використайте з набору фіксованих в конфігурації ПЛК, наприклад *DO4*. Поточне значення змінної *rez* на аналоговому вході має тип *REAL*. Але для визначення часу імпульсу повинна бути використана змінна типу *TIME*. Тому в програмі потрібно передбачити перетворення значення змінної з типу *REAL* на тип *TIME*. Нехай період слідування імпульсів буде дорівнювати

1000мс. Тому час імпульсу буде відповідати перетвореному значенню резистора, а час паузи буде дорівнювати значенню **1000мс-рез.** Також в програмі можна використати обмеження мінімального та максимального часу імпульсу. Заповніть робоче поле **POU PLC_PRG** згідно з екранною формою, що зображена на рис. 3.7.

У результаті розроблено проект з можливістю ручного встановлення тривалості імпульсу при ШІМ-регулюванні. Зробіть візуалізацію проекту, в якій створіть тренди для відображення значення вхідного та вихідного параметрів на одному графіку. Подайте проект викладачу для перевірки. Побудуйте графік залежності тривалості імпульсу при ШІМ-регулюванні на дискретному виході від значення параметра на аналоговому вході ПЛК.

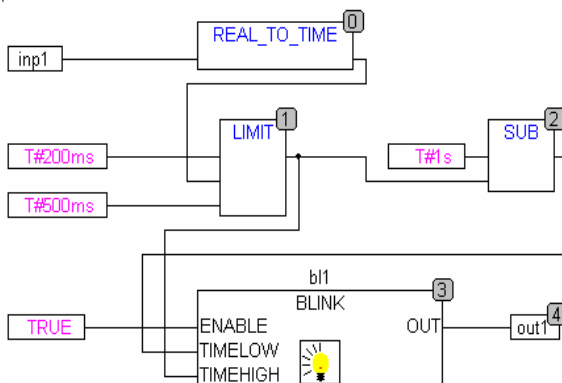


Рисунок 3.7 – Програма ШІМ управління вихідним каналом

3.4. Перевірка роботи макета системи управління

1) Перевірте правильність роботи проектів з розробленими програмами управління. Для цього змінійте опір резистору та вводіть значення локальних змінних та спостерігайте за всіма змінами на екрані монітора ПК. Подайте проекти викладачу для перевірки.

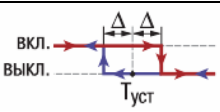
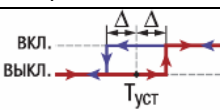
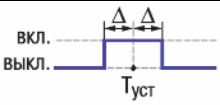
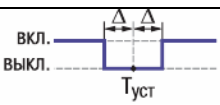
2) Складіть звіт в редакторі *Microsoft Office* відповідно до правил оформлення звітів: відомості про виконавця, назву та мету роботи, схему з'єднань, лістинги програм ПЛК, параметри налаштувань COM-порту для завантаження проектів.

3.5. Завдання для самостійної роботи

1) Розробіть проект, що реалізує дискретне управління нагрівачем або охолоджувачем залежно від поточного значення температури.

Необхідно реалізувати в проекті функції згідно з табл. 3.1. Логіка роботи регулятора вибирається відповідно до номера варіанта. Додатково в проекті передбачте можливість встановлення гістерезису (10 % від всього діапазону) та можливість затримки перемикання вихідного пристрою на три секунди на випадок збурень та похибок при вимірюванні параметра. Зробіть візуалізацію проекту.

Таблиця 3.1 Варіанти умов для вмикання вихідних елементів

№ з/п згідно з журн.	Тип логіки регулятора	<i>LAMP</i> <i>_TEN</i>	<i>LAMP</i> <i>_VENT</i>	<i>TEN</i>	<i>VENT</i>
1, 5, 9, 13, 17, 21, 25	 «нагрівач»	<i>DO1</i>	-	+	-
2, 6, 10, 14, 18, 22	 «охолоджувач»	-	<i>DO2</i>	-	+
№ з/п згідно журн.	Тип логіки сигналізатора	<i>S_OUT</i>		<i>S_IN</i>	
3, 7, 11, 15, 19, 23	 П-подібний сигналізатор	<i>DO2</i>		<i>DO1</i>	
4, 8, 12, 16, 20, 24	 U-подібний сигналізатор	<i>DO1</i>		<i>DO2</i>	

В якості датчика виберіть змінний опір, що імітує реальний термометр опору. У програмі має бути реалізоване реальне управління вихідними пристроями (нагрівачем – *TEN* або охолоджувачем – *VENT*). Також повинна здійснюватися світлова сигналізація, що показує стан вихідних пристроїв (*LAMP_TEN* і *LAMP_VENT*) або вхід/вихід контролюваного параметра за вказані межі, – *S_IN* або *S_OUT*.

В табл. 3.1 рисунки відповідають таким типам регуляторів:

- «нагрівач», тобто 2-позиційний регулятор «прямий гістерезис»;
- «охолоджувач», тобто 2-позиційний регулятор «зворотний гістерезис»;
- П-подібний сигналізатор, тобто вихід спрацьовує при вході параметра в межі;
- U-подібний сигналізатор, тобто вихід спрацьовує при виході параметра за межі.

На рисунках, наведених в табл. 3.1 прийняті такі позначення: $T_{уст}$ – значення уставки (температури), Δ – значення гістерезису.

2) Розробіть проект 2-позиційного регулятора з використанням ФБ *ON_OFF_HIST_REG* з бібліотеки *Util.lib*. Нехай уставка буде дорівнювати 35 °C, гістерезис – 5 °C та зона нечутливості – 1 °C.

3.6. Контрольні питання

- 1) Поясніть принцип дії 2-позиційного регулятора параметра.
- 2) Наведіть загальну схему роботи регулятора параметра.
- 3) Дайте коротку характеристику принципу дії ШІМ-регулятора.
- 4) Чим відрізняється апаратний ШІМ-регулятор від програмного?
- 5) Дайте коротку характеристику бібліотеки *Util.lib*.
- 6) Яким чином працюють сигналізатори параметра?
- 7) Наведіть алгоритм роботи блока *BLINK* з бібліотеки *Util.lib*.
- 8) Як визначається параметр шпаруватості в апаратному ШІМ-регуляторі?
- 9) Як визначається параметр шпаруватості в програмному ШІМ-регуляторі?

ПРИНЦИПИ РЕАЛІЗАЦІЇ ПІД-РЕГУЛЮВАННЯ

4.1. Мета роботи

- закріплення теоретичних знань щодо принципів реалізації ПІД-регулювання;
- розробка програм користувача для ПІД-регулювання на прикладі використання стандартних функціональних блоків з бібліотек *UTIL.lib*, *PID_Rerulators.lib* та *pid_reg2.lib*.

4.2. Порядок виконання роботи

Виконання лабораторної роботи складається з наступних етапів:

- 1) Створення проекту для ПІД-регулювання 2-позиційним виконавчим пристроєм з використанням ШІМ-управління дискретним виходом.
- 2) Створення проекту ПІД-управління 2-позиційним виконавчим пристроєм з автоналаштуванням до параметрів об'єкта регулювання.
- 3) ПІД-управління 3-позиційним виконавчим механізмом з регульовальним органом типу «клапан» та автоналаштуванням до параметрів об'єкта.

4.3. Хід виконання роботи

4.3.1. Короткі відомості про ПІД-регулювання

Здійснення ПІД-регулювання при управлінні технологічними процесами в середовищі *CoDeSys* реалізується за допомогою додаткових бібліотек. Це, наприклад, бібліотека *UTIL.lib* з папкою *controller*, яка входить до інсталяційного пакета *CoDeSys*. До цієї папки входить три ФБ: ПІД-, ПІД-регулятори та регулятор з фіксованим часом циклу. Крім того, розробники ПЛК на платформі *CoDeSys* пропонують власні бібліотеки для своїх ПЛК. Так, компанія ОВЕН разом з стандартним набором програмного забезпечення *CoDeSys* надає додаткові безкоштовні бібліотеки для реалізації ПІД-регулювання та управління різними виконавчими механізмами: *PID_Rerulators.lib* та *pid_reg2.lib*. Але особливістю цих бібліотек є неможливість виконання ФБ бібліотеки в емуляторі ПЛК, тобто для налагодження програм користувача потрібна обов'язкова наявність реального ПЛК. Крім того, ці бібліотеки не працюють в ПЛК

інших виробників. Нагадаємо, що основне завдання ФБ з цих бібліотек – це підтримання поточного параметра (*PV*) на заданому рівні (*SP*) шляхом впливу на виконавчі механізми. Загальна схема ПІД-регулятора зображена на рис. 4.1.

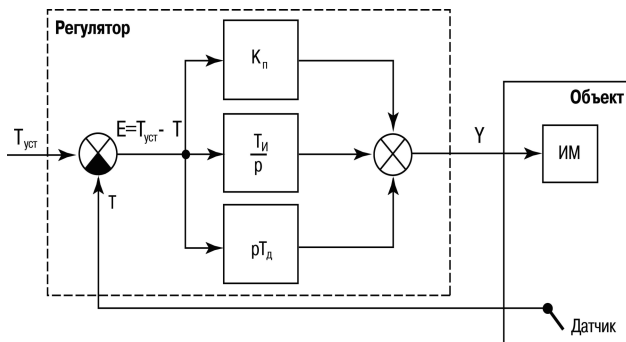


Рисунок 4.1 – Загальна схема ПІД-регулятора

Пропорційно-інтегрально-диференціальний (ПІД) регулятор використовується в системах автоматики для підтримки з високою точністю потрібних параметрів. Він видає вихідний сигнал, направлений на зменшення відхилення поточного значення регульованої величини від уставки (завдання). У загальному випадку робота універсального ПІД-регулятора для вихідного сигналу (Y_i) може бути описана рівнянням

$$Y_i = \frac{1}{X_p} \cdot \left[E_i + \tau_a \frac{\Delta E_i}{\Delta t_{\dot{e}ci}} + \frac{1}{\tau_e} \sum_{i=0}^n E_i \Delta t_{\dot{e}ci} \right],$$

де X_p – смуга пропорційності; \dot{A}_s – різниця між уставкою і поточним значенням контролюваної величини або розузгодження; ΔE_i – різниця між двома послідовними вимірами \dot{A}_s та \dot{A}_{s-1} ; $\Delta t_{\dot{e}ci}$ – час між двома послідовними вимірами; τ_a – постійна диференціювання; τ_e – постійна інтеграції; $\sum_{i=0}^n E_i$ – накопичена сума розузгодження.

4.3.2. Використання ФБ з бібліотеки *UTIL.lib* для ПД-регулювання

Для автоматичної підтримки параметра на заданому рівні (наприклад, температури, тиску тощо), можуть використовуватися будь-які модифікації контролерів, що мають аналоговий вхід для вимірювання параметра та дискретний вихід для ШІМ-управління 2-позиційним виконавчим пристроєм або аналоговий вихід для управління за допомогою електропневмоперетворювача (ЕПП) мембранним виконавчим механізмом (МІМ). Розглянемо перший варіант: аналоговий вхід/дискретний вихід.

Завдання для програмування: у сушильній шафі необхідно підтримувати певну температуру. Вибір потрібного значення температури (+40 °C або +50 °C) та перемикання режиму проводиться оператором. Система повинна з максимально можливою швидкістю виходити на заданий режим.

Отже, створіть проект з ім'ям *lr_4_1_name.pro* та підключить бібліотеку *UTIL.lib* до нього. В якості цільової платформи виберіть *ОВЕН ПЛК150.I-L*. Мову реалізації *POU PLC_PRG* оберіть *CFC*. На рис. 4.2 наведено принципіальну електричну схему системи регулювання температури. На схемі показано термометр опору та нагрівач, які підключені до входу та виходу відповідно.

Проведіть конфігурування ресурсів ПЛК відповідно до схеми системи регулювання (див.

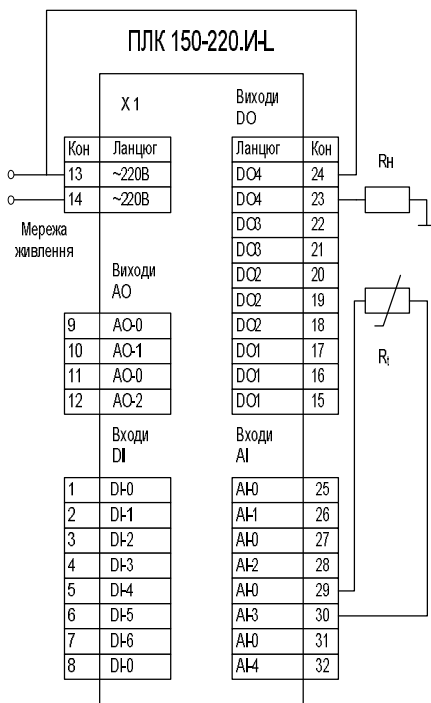


Рисунок 4.2 – Схема електричних з'єднань ПЛК з зовнішніми пристроями

рис. 4.2). При цьому необхідно передбачити можливість вибору потрібного значення уставки за допомогою перемикача типу «сухий контакт». На рис. 4.3 зображений фрагмент програми користувача з ПІД-регулятором в режимі «нагрівача».

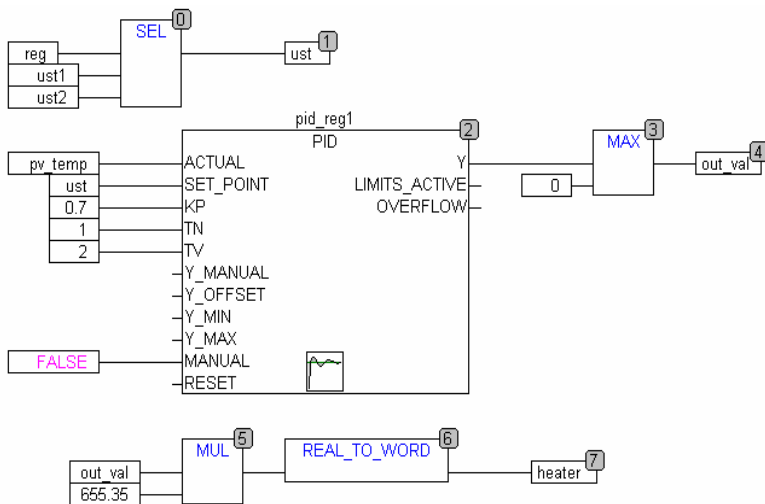


Рисунок 4.3 – Фрагмент програми користувача

У проєкті за допомогою програмних блоків 3, 5 та 6 (див. рис. 4.3) розрахований в блоці ПІД-регулятора сигнал управління перетворюється в значення параметра програмного ШІМ-модулятора на відповідному дискретному виході. При цьому спочатку відкидаються негативні значення сигналу, потім здійснюється його масштабування з метою лінійного приведення сигналу діапазону (0...100 %) до діапазону ШІМ-регулятора (0...65535 одиниць) і, насамкінець, перетворення змінної типу *REAL* на тип *WORD*.

Аналогічним чином може бути розроблений проєкт з ПІД-регулятором в режимі «холодильника». Управляючий сигнал також необхідно перетворити, але відкинувши його позитивні значення.

Розробить візуалізацію проєкту з відображенням трендів з поточним параметром, потужністю управляючого сигналу та завданням. У візуа-

лізації створіть графічний елемент для вводу потрібного завдання та параметрів регулятора. Подайте проект викладачу для перевірки.

4.3.3. Порядок використання ФБ з бібліотеки *PID_Rerulators.lib*

Наступне завдання аналогічне попередньому, але використайте ФБ для ПД-управління 2-позиційним механізмом («нагрівачем» або «охолоджувачем») з автоналаштуванням параметрів регулятора по відношенню до об'єкта регулювання. Тому використайте схему, що зображена на рис. 4.2. Далі створіть проект з ім'ям *lr_4_2_name.pro* та підключить бібліотеку *PID_Rerulators.lib* до нього. В якості цільової платформи виберіть *OBEH ПЛК150.I-L*. Мову реалізації *POU PLC_PRG* оберіть *CFC*.

На рис. 4.4 зображений фрагмент програми користувача для ПД-регулювання за допомогою 2-позиційного ВМ та можливістю вмикання та вимикання режиму автоналаштування ПД-регулятора.

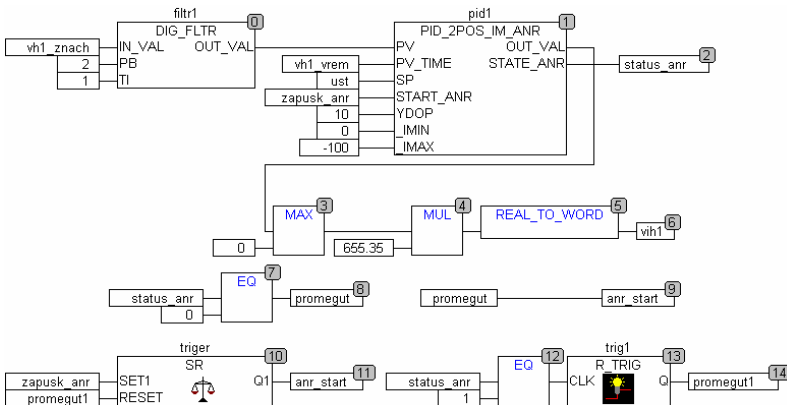


Рисунок 4.4 – Фрагмент програми користувача

На рис. 4.5 наведено фрагмент програми користувача з розділом оголошених змінних.

Перші чотири змінних різних типів зв'язані з відповідними фізичними входами та виходами ПЛК. Змінні, що залишилися, є локальними, і потрібні для функціонування програми користувача та контролю за режимом роботи ПД-регулятора.

```

0001 PROGRAM PLC_PRG
0002 VAR
0003   vh1_znach AT%ID4.0 : REAL; (*Измеренное значение на аналоговом входе №1*)
0004   vh1_vrem AT%IW4.1 : WORD; (*Измеренное время на аналоговом входе №1*)
0005   запуск_anr AT%IX0.0 : BOOL; (*Дистанционный запуск АНР с помощью кнопки на дискретном входе №1*)
0006   vih1 AT%QW1.0.0 : WORD; (*к дискретному выходу №1 подключен подмодуль ШИМ*)
0007   filtr1 : DIG_FLTR; (*Цифровой фильтр на измеренное значение*)
0008   ust : REAL := 50; (*Уставка, с предустановкой 50*)
0009   anr_start : BOOL := TRUE; (*Команда запуска автонастройки. При первом запуске включается автоматически*)
0010   status_anr : BYTE; (*Статус прохождения автонастройки*)
0011   trigger : SR;
0012   promegut : BOOL;
0013   promegut1 : BOOL;
0014   trig1 : R_TRIG;
0015   pid1 : PID_2POS_IM_ANR; (*Блок ПИД с автонастройкой*)
0016 END_VAR

```

Рисунок 4.5 – Фрагмент програми користувача з оголошеними змінними

Розробіть візуалізацію проекту з відображенням трендів з поточним параметром та значенням розрахованого управляючого параметра. Подайте проект викладачу для перевірки.

4.3.4. Створення проекту ПІД-регулятором бібліотеки *pid_reg2.lib*

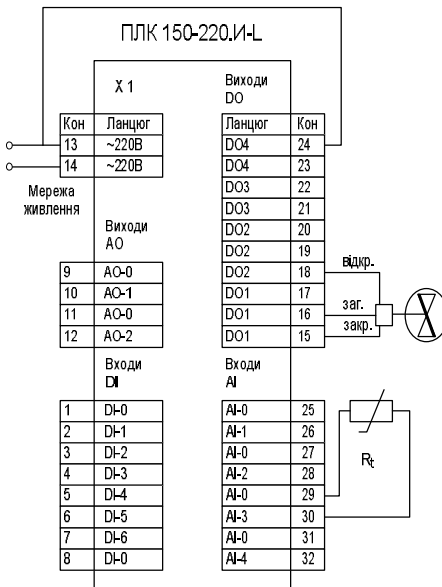


Рисунок 4.6 – Схема електричних з'єднань ПЛК з зовнішніми пристроями

На відміну від перших двох проектів, де реалізовано ПІД-управління 2-позиційним ВМ (ТЕНом), в наступному буде розглянута можливість ПІД-управління 3-позиційним ВМ з регульовальним органом типу «клапан» та автоналаштуванням до параметрів об'єкта.

Отже, створіть проект з ім'ям *lr_4_3_name.pro* та підключить бібліотеку *pid_reg2.lib* до нього. В якості цільової платформи виберіть *ОВЕН ПЛК150.I-L*. Мову реалізації *POU PLC_PRG* оберіть *CFC*.

На рис. 4.6 зображена схема електричних з'єднань ПЛК з зовнішніми пристроями. Відзначимо, що для реалізації режимів управління ВМ «більше» та «менше» в схемі використано два перших дискретних виходи ПЛК, причому сигнальні лінії підключені до нормально-закритих контактів перемикаючих входів. Заповнить поле оголошення змінних згідно з екранною формою, що зображена на рис. 4.7. Перші п'ять змінних різних типів зв'язані з відповідними фізичними входами та виходами ПЛК. Змінні, що залишилися, є локальними, і потрібні для функціонування програми користувача та контролю за режимом роботи ПІД-регулятора.

```

0001 PROGRAM PLC_PRG
0002 VAR
0003   vh_znach AT%I4.0 : REAL;    (*Измеренное значение на аналоговом входе №1*)
0004   vh_vrem AT%IW4.1 : WORD;    (*Измеренное время на аналоговом входе №1*)
0005   anr_start AT%X0.0 : BOOL;   (*Дистанционный запуск АНР с помощью кнопки на дискретном входе №1*)
0006   vih1 AT%X1.0 : BOOL;       (*к дискретному выходу №1*)
0007   vih2 AT%X1.1 : BOOL;       (*к дискретному выходу №2*)
0008   fltr1: DIG_FLTR;           (*Цифровой фильтр на измеренное значение*)
0009   ust: REAL := 40;           (*Уставка, с предустановкой 40*)
0010   wem_hod: REAL := 90;       (*время хода задвижки, в зависимости от характеристик ИМ*)
0011   reg_1: APID_VALVE;         (*ПИД регулятор с автонастройкой для трехходовых ИМ без датчика положения*)
0012   nach_moshn: REAL := 20;   (*Начальное значение мощности при автонастройке*)
0013   zona_nechustv: REAL := 5; (*Зона нечувствительности в ед. параметра*)
0014   luf: REAL := 0.5;         (*Время выборки люфта при реверсе, в секундах*)
0015   sinhro: BOOL;             (*Задавать положение задвижки из мат. модели *)
0016   moshn_upr: REAL;
0017   pr_anr: BOOL;
0018   pr_bv_ust: BOOL;
0019 END_VAR

```

Рисунок 4.7 – Фрагмент програми користувача з оголошеними змінними

Фрагмент програми користувача, яка використовує ФБ *APID_VALVE* з бібліотеки *pid_reg2.lib*, показаний на рис. 4.8. У цьому проекті також використаний ФБ *DIG_FLTR* з бібліотеки *PID_Rerulators.lib*, який зменшує вплив випадкових імпульсних перешкод на вимірюваний параметр. Вхідні параметри для ФБ встановлені в

розділі оголошення змінних і є актуальними одночасно з запуском проекту на ПЛК.

Розробить візуалізацію проекту з відображенням трендів з поточним параметром та значенням розрахованої потужності управляючого параметра. Також створить індикатори режимів управління ВМ «більше» та «менше» Подайте проект викладачу для перевірки.

4.4. Перевірка роботи макета системи управління

1) Перевірте правильність роботи проектів з розробленими програмами управління. Для цього вмикайте/вимикайте перемикачі, змінійте параметри налаштувань ФБ, вводіть значення локальних змінних та спостерігайте за всіма змінами на екрані монітора ПК. Подайте проекти викладачу для перевірки.

2) Складіть звіт в редакторі *Microsoft Office* відповідно до правил оформлення звітів: відомості про виконавця, назву та мету роботи, схему з'єднань, лістинги програм ПЛК, параметри налаштувань COM-порту для завантаження проектів.

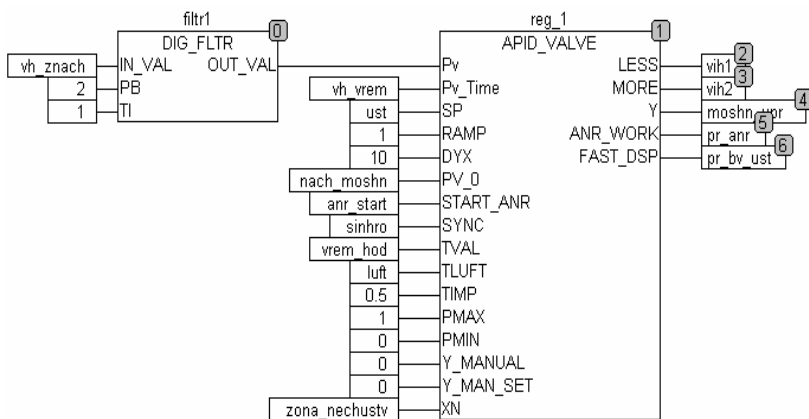


Рисунок 4.8 – Фрагмент програми користувача

4.5. Завдання для самостійної роботи

1) Розробить проект ПІД-регулятора на основі ФБ *PID* з використанням двох виконавчих пристроїв: «нагрівача» та «охолоджувача», які

підключені до різних дискретних виходів ПЛК з програмним ШІМ-модулятором.

2) Розробіть проект ПД-регулятора з використанням ФБ *PID_2POS_IM_ANR* для управління 2-позиційним виконавчим пристроєм, який підключений до аналогового виходу 4...20 мА.

3) Розробіть проект ПД-регулятора з використанням ФБ *APID_PWM* для управління 2-позиційним виконавчим пристроєм – «нагрівачем» або «охолоджувачем», які підключені до дискретних виходів з програмним ШІМ-модулятором.

4.6. Контрольні питання

- 1) Дайте коротку характеристику принципів ПД-регулювання.
- 2) Перелічіть доступні бібліотеки з ФБ для реалізації ПД-регулювання за допомогою ПЛК *ОВЕН*.
- 3) Яким чином можна реалізувати ПД-регулювання за допомогою дискретних входів ПЛК?
- 4) Яким чином здійснюється фільтрація значень параметра?
- 5) Як враховуються характеристики виконавчих пристроїв під час управління клапаном без датчика положення та з датчиком положення?

СПИСОК ЛІТЕРАТУРИ

1. Петров И. В. Программируемые контроллеры. Стандартные языки и приемы программирования / И.В. Петров. – М. : СОЛОН-ПРЕСС, 2008. – 256 с.
2. Руководство пользователя по программированию ПЛК в CoDeSys V2.3. – Смоленск : ПК Пролог, 2005, 453 с.
3. Контроллер программируемый логический ОВЕН ПЛК150. Паспорт и Руководство по эксплуатации – М. : ОВЕН, 98 с.
4. Конфигурирование области ввода/вывода ПЛК. Руководство пользователя – М. : ОВЕН, 119 с.
5. Ел. джерело: <http://www.3S-software.com>.
6. Ел. джерело: <http://www.owen.ru>.

ЗМІСТ

Вступ	3
<i>Лабораторна робота №1. Загальні принципи програмування контролерів OВЕН в середовищі CODESYS</i>	<i>4</i>
<i>Лабораторна робота №2. Структурування проектів та основні принципи використання таймерів та лічильників</i>	<i>27</i>
<i>Лабораторна робота №3. Принципи реалізації 2-позиційного та ШІМ регулювання.</i>	<i>34</i>
<i>Лабораторна робота №4. Принципи реалізації ПІД-регулювання . . .</i>	<i>45</i>
Список літератури	53

[illegible]

Навчальне видання

МЕТОДИЧНІ ВКАЗІВКИ

для проведення лабораторних занять з курсу
«Програмне забезпечення мікропроцесорних систем»
для студентів напряму підготовки 050202
«Автоматизація та комп'ютерно-інтегровані технології»

Укладачі: ТОШИНСЬКИЙ Володимир Ілліч
ЛИСАЧЕНКО Ігор Григорович
ЛИТВИНЕНКО Ігор Іванович
ШУТИНСЬКИЙ Олексій Григорович
ДЗЕВОЧКО Олександр Михайлович
ГЕРМАН Едуард Євгенович

Відповідальний за випуск В. І. Тошинський
Роботу до видання рекомендувала Н. М. Самойленко

Редактор Л. Л. Яковлева

План 2012 р., поз. 80

Підп. до друку __. __. 2012. Формат 60×84 1/16. Папір офсетний.
Riso-друк. Гарнітура Таймс. Ум. друк. арк. 2,8.
Наклад 100 прим. Зам. № _____. Ціна договірна.

Видавничий центр НТУ „ХП”.

Свідоцтво про державну реєстрацію ДК № 3657 від 24.12.2009 р.
61002, Харків, вул. Фрунзе, 21.

ТОВ "Видавництво "Підручник НТУ "ХП".

Свідоцтво про державну реєстрацію ДК № 3656 від 24.12.2009 р.
61002, Харків, вул. Фрунзе, 21.